FPGA-accelerated machine learning inference as a solution for particle physics computing challenges

Jennifer Ngadiuba, Maurizio Pierini **(CERN)**
Javier Duarte, Burt Holzman, Ben Kreis, Kevin Pedro, ***Mia Liu***, Nhan Tran, Aristeidis Tsaris **(Fermilab)**
Phil Harris, Dylan Rankin **(MIT)**
Zhenbin Wu **(UIC)**

Dec 9, CPAD, 2018

OUTLINE

Motivation

Challenges of big science and computing

Our solution : proof of concept

Particle physics computing with Brainwave

Physics cases

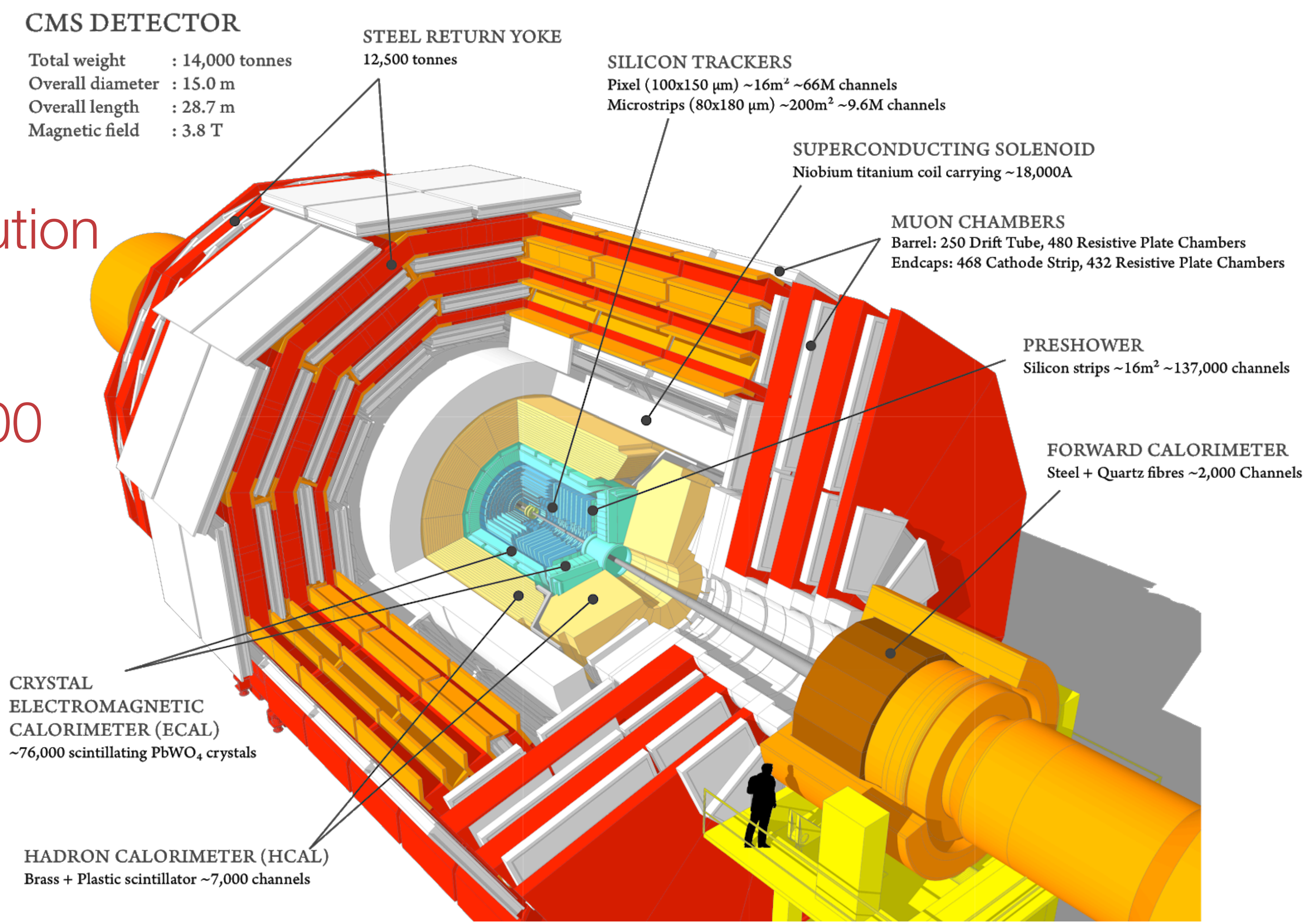
NOvA & jet identification at collider experiments

Outlook& takeaways

Motivation: Challenges of big science and computing

CMS as an example: Detectors becoming increasingly complex

- High-resolution detector
- Order of 100 Million channels



DETECTORS GETTING MORE COMPLEX!

CMS upgrade to get ready for HL-LHC data-taking: higher granularity, timing information etc.

Example: CMS High Granularity Calorimeter

Total Silicon:

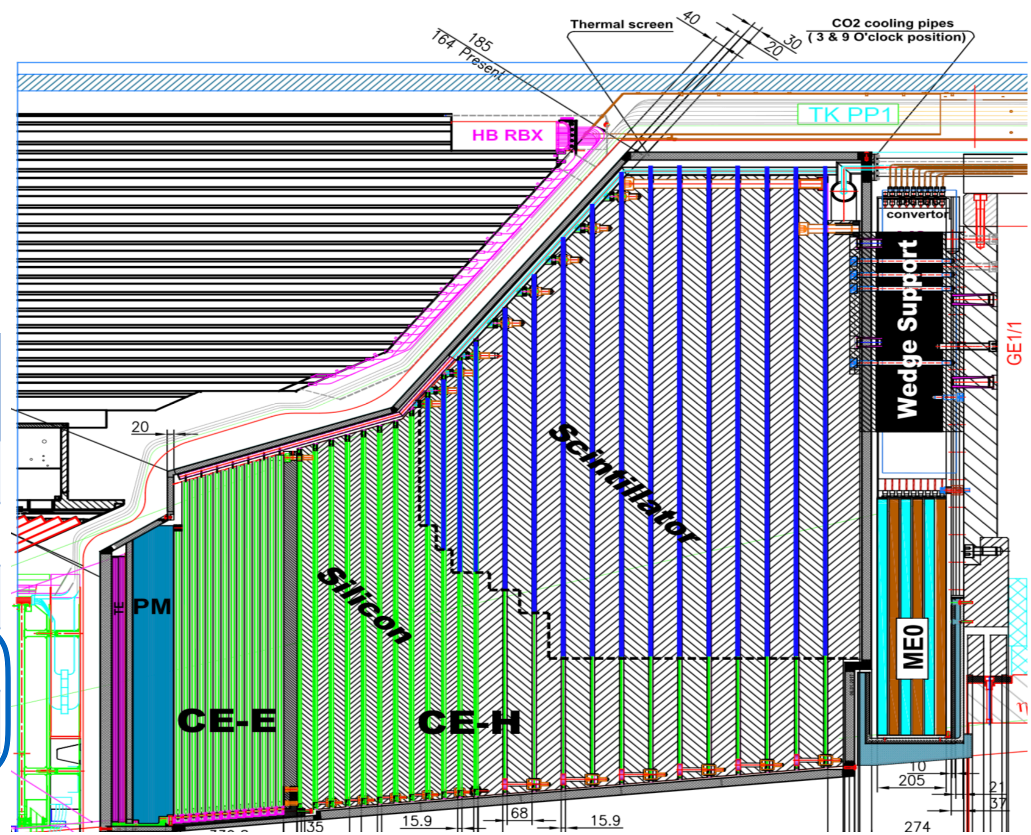
- 600 m^2

Total scintillator

- 500 m^2

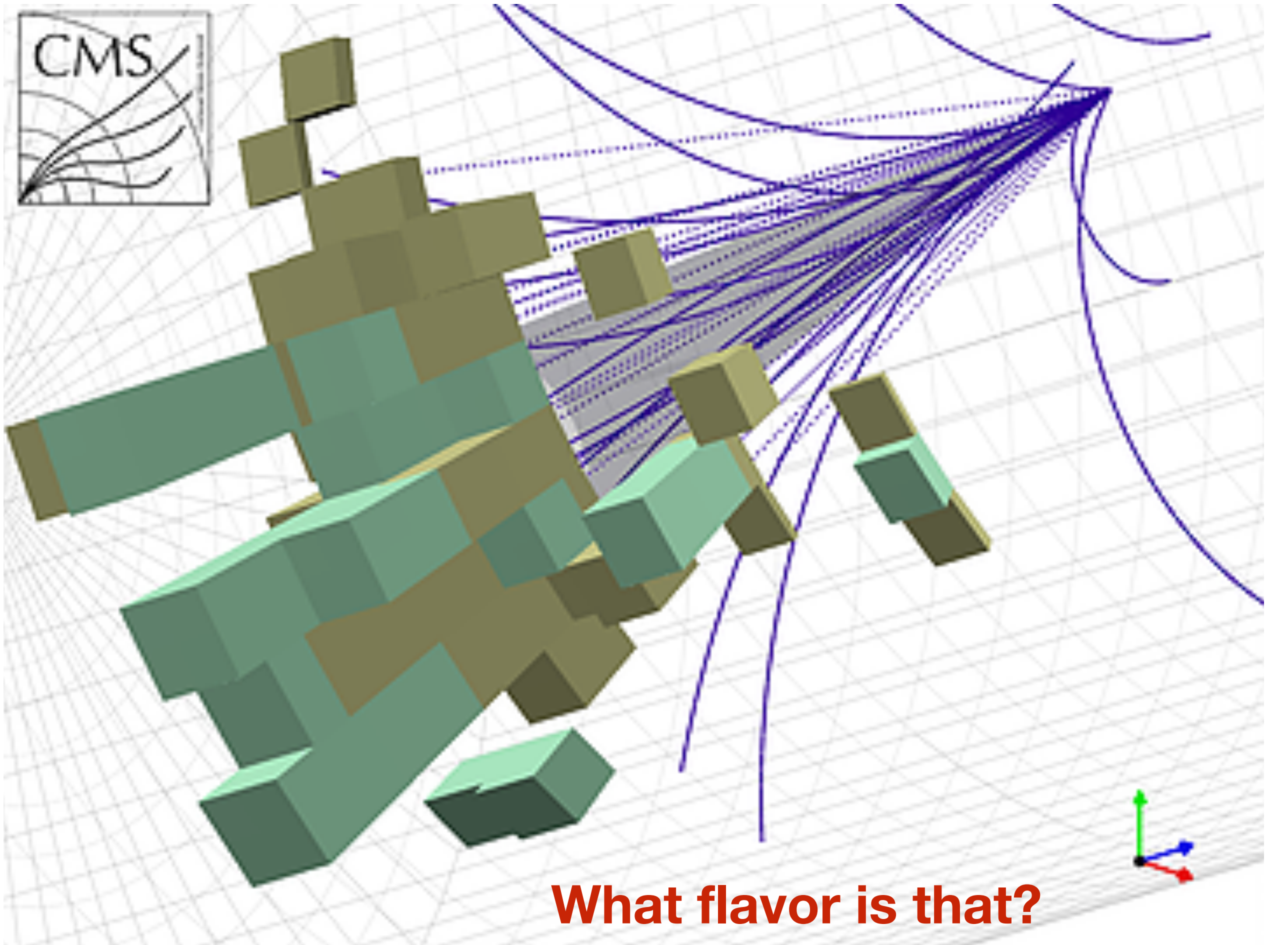
	CMS	ATLAS	CMS HGCal
Diameter (m)	15	25	
Length (m)	28.7	46	
B-Field (T)	3.8	2/4	
EM Cal channels	~80,000	~110,000	4.3M
Had Cal channels	~7,000	~10,000	1.8M

P.Merkel



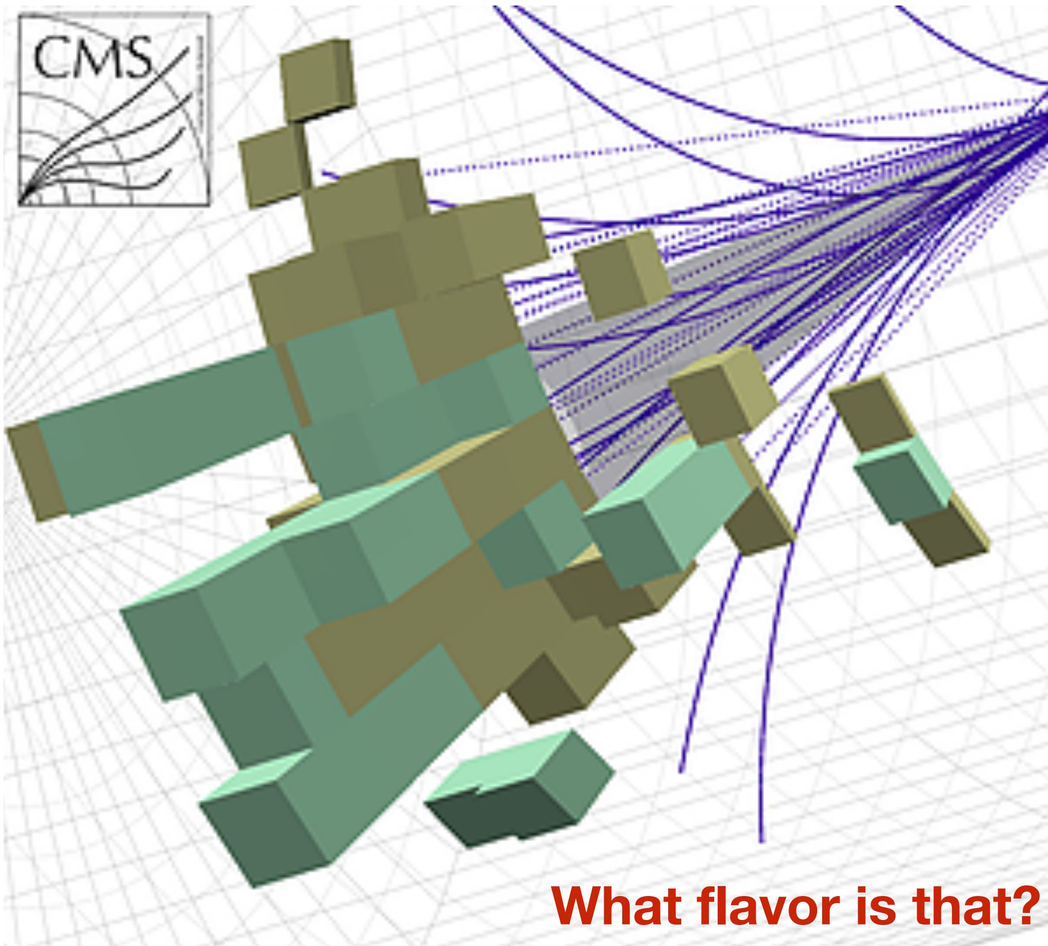
SOPHISTICATED ALGORITHMS

CMS as an example: Need sophisticated algorithms to fully exploit the information taken by more complex detectors

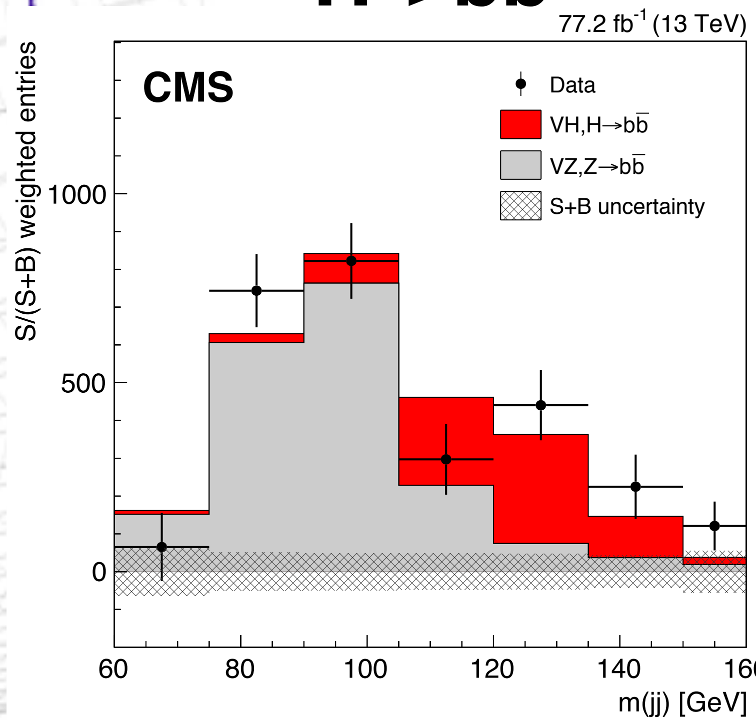


CRITICAL FOR DISCOVERIES

CMS as an example: plenty of physics cases

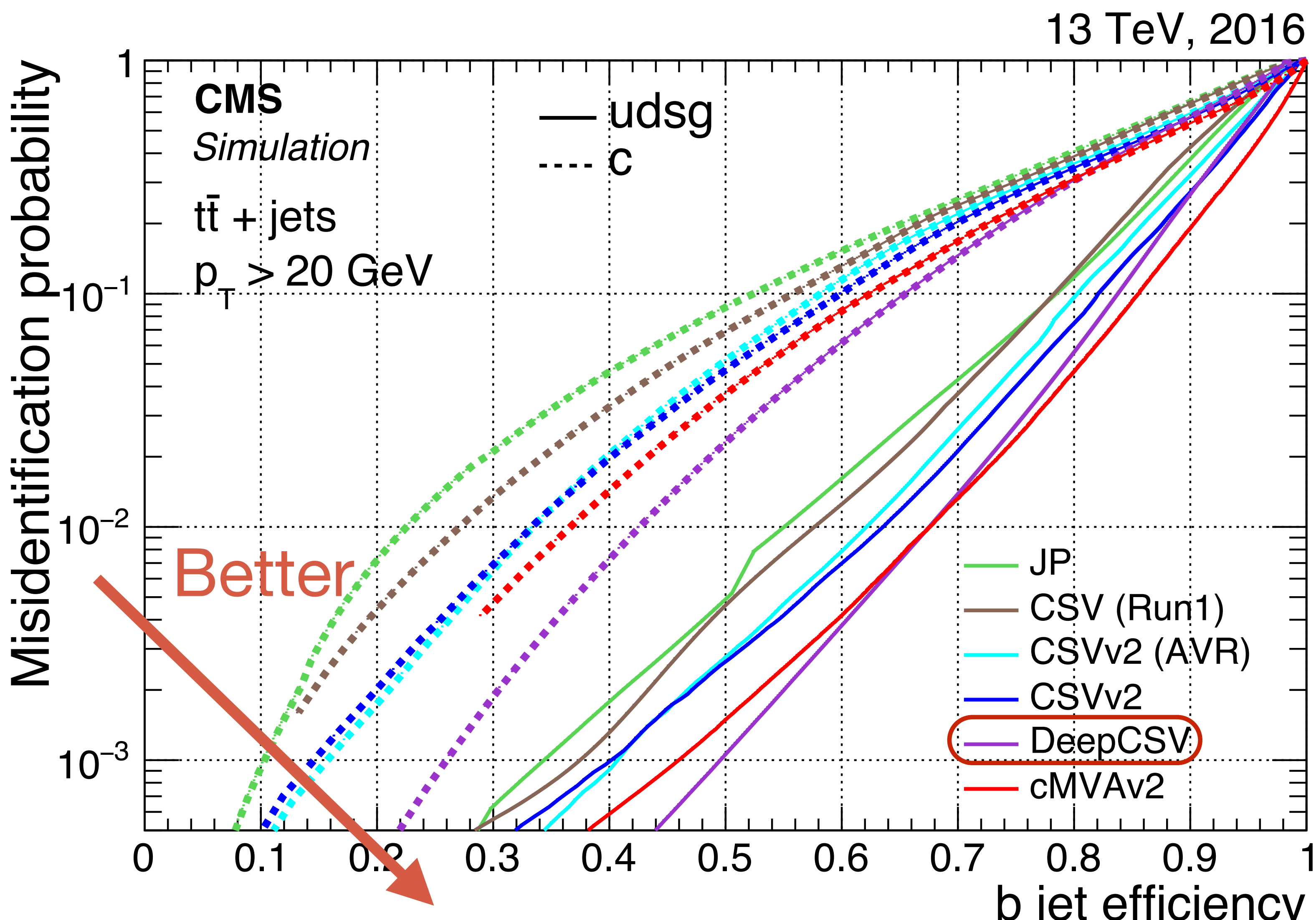


Yukawa coupling: H->bb



BOOM IN USING DEEP NEURAL NETWORKS

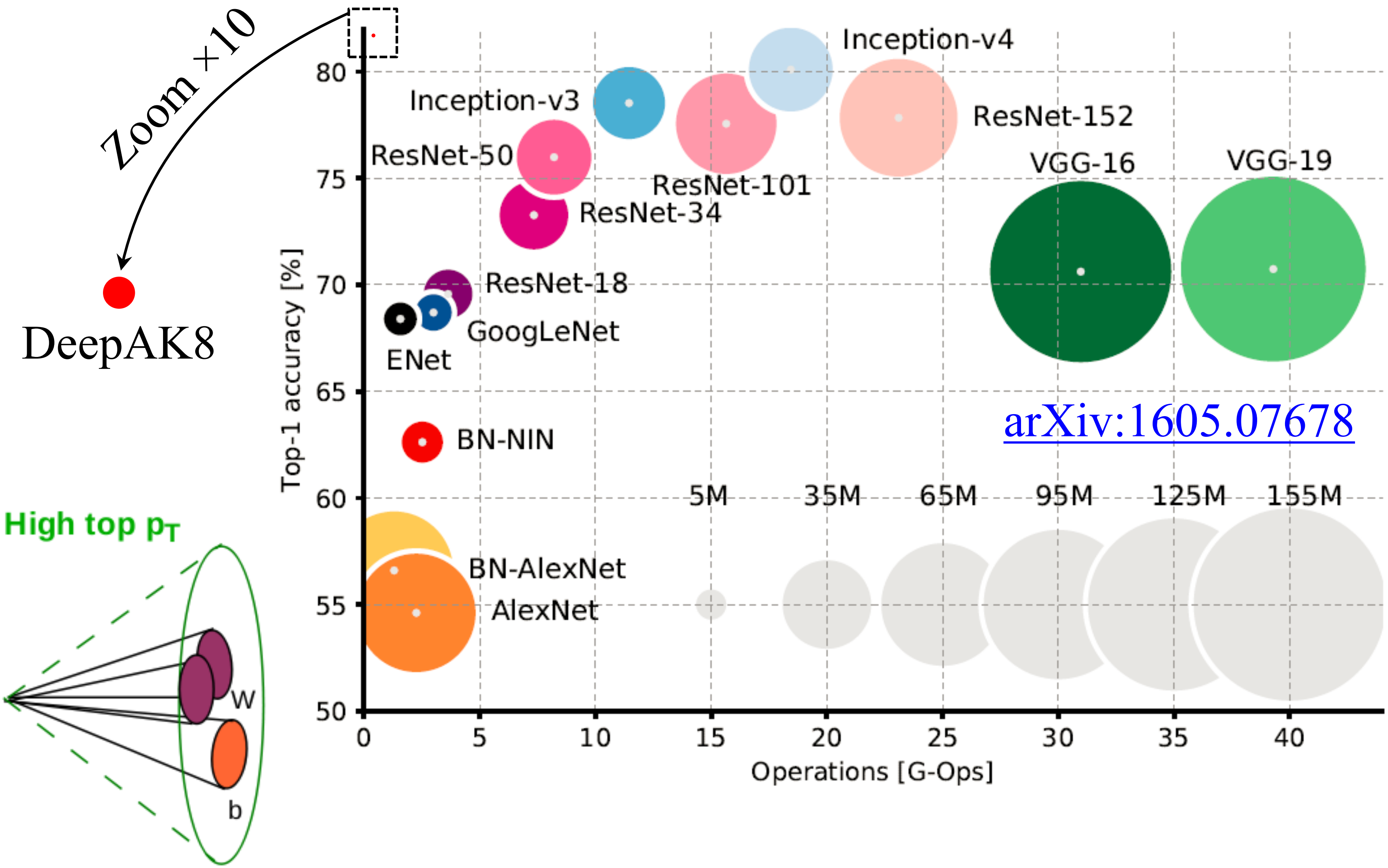
Deep neural network based algorithms perform the best



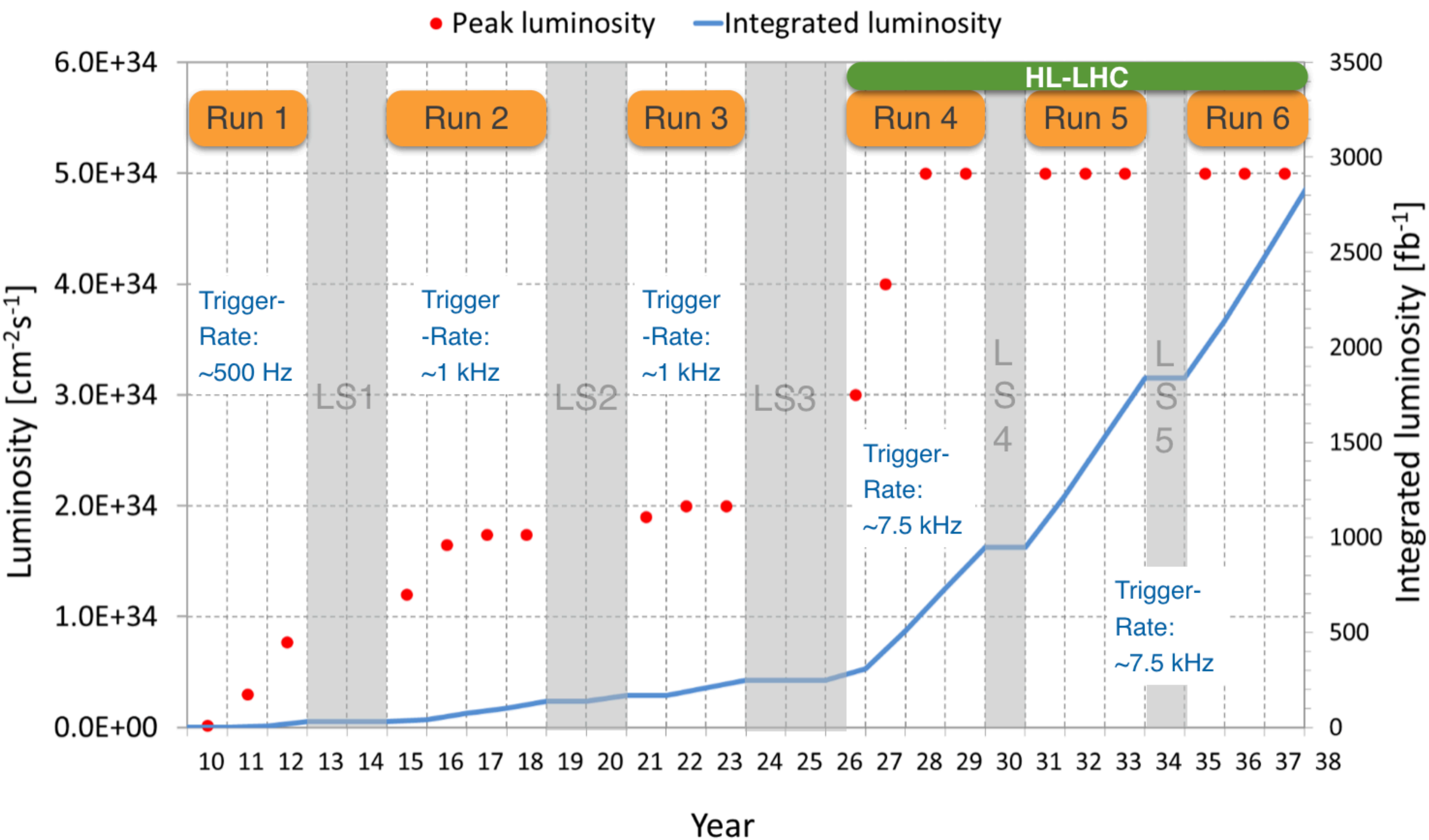
EVENT COMPLEXITY WILL GROW

Networks can grow bigger, number of networks will increase

Network inferencing taking significant fraction of the final event processing time in CMS

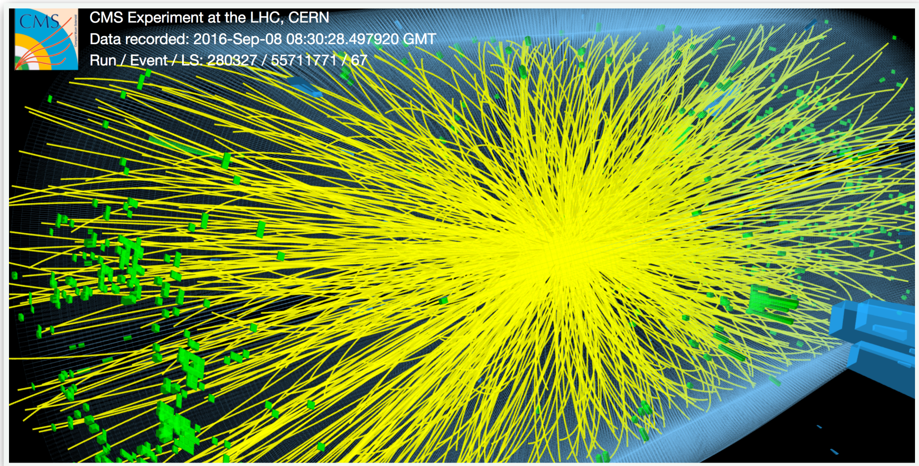


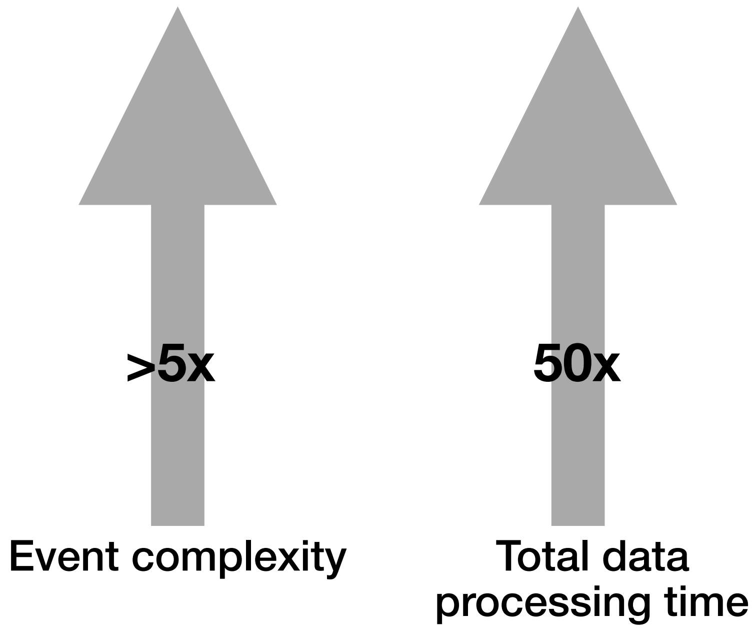
GROWING DATASET

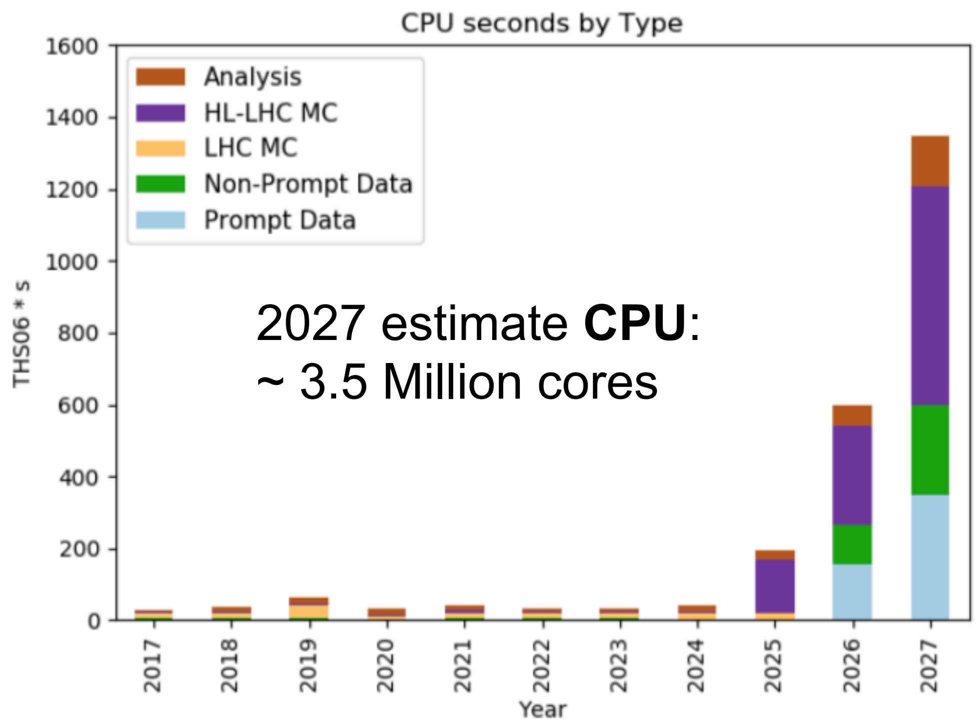


All factors: The computing challenge

Current: ~5 minutes per HL-LHC event

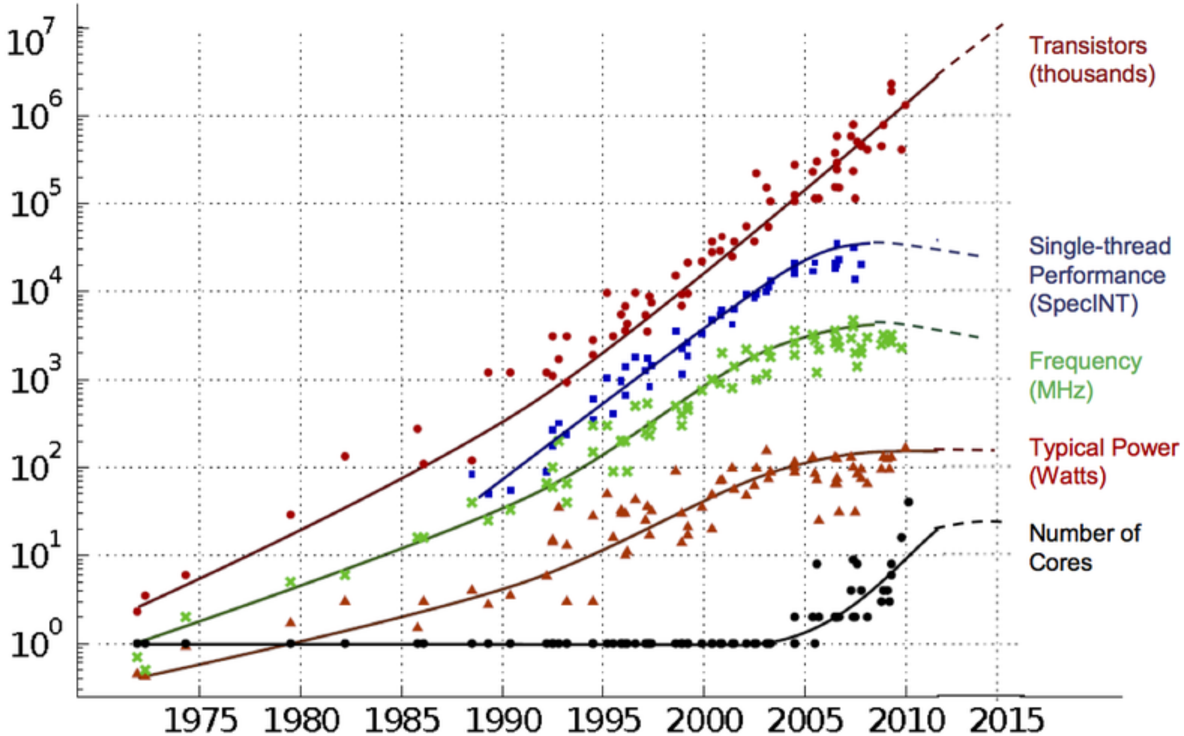






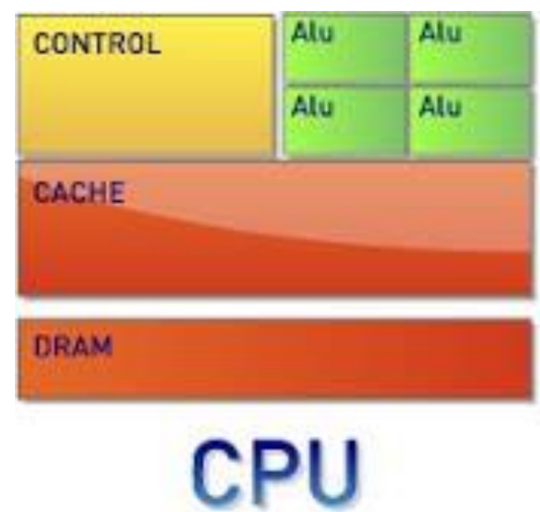
2027 estimate **CPU**:
~ 3.5 Million cores

MOORE'S LAW AND DENNARD SCALING



Original data collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond and C. Batten
Dotted line extrapolations by C. Moore

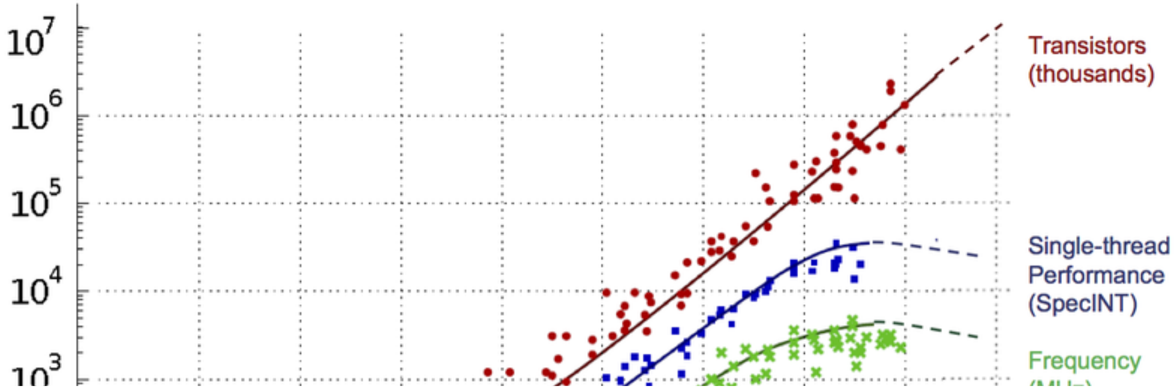
Moore's Law continues
…but Dennard Scaling fails



Single threaded performance not improving

Circa ~2005: "The Era of Multicore"

MOORE'S LAW AND DENNARD SCALING



Moore's Law continues
…but Dennard Scaling fails

We are not the only one facing the computing challenges faced with AI boom and data volume explosion

Single threaded performance not improving

Circa ~2005: "The Era of Multicore"

→ **Today: Transition to the "Era of Specialization"?** (c.f. Doug Burger)

CMS PARTY@2016

Remember that Facebook ask you (at least used to) to tag people when you upload a photo?

CMS PARTY@2016

Runs image detection every time some one uploads a photo:
Neutral network inference

CMS PARTY@2016

300 million photos uploaded/day as of 2018.Nov

Platform Pros & Cons:

More Flexible

Homogeneous

Specialized

More Efficient

	Perf/W			
CPUs	1X	Today's standard, most programmable, good for services changing rapidly	Conventional programming	C/C++
Manycore CPUs	3X	Many simple cores (10s to 100s per chip), useful if software can be fine-grain parallel, difficult to maintain.		
GPUs	5-30X	Good for data parallelism by merged threads (SIMD), High memory bandwidth, power hungry	Alternative programming	CUDA
FPGAs	5-30X	Most radical fully programmable option. Good for streaming/irregular parallelism. Power efficient but currently need to program in H/W languages.		Verilog
Structured ASICS	20-100X	Lower-NRE ASICs with lower performance/efficiency. Includes domain-specific (programmable) accelerators.	Can't change functionality	Verilog
Custom ASICs	> 100X	Highest efficiency. Highest NRE costs. Requires high volume. Good for functions in very widespread use that are stable for many years.		

PLATFORM PROS & CONS FOR INDUSTRY:

More Flexible

Homogeneous

Platform	Perf/W	Pros & cons
CPUs	1X	Today's standard, most programmable, good for services changing rapidly
Manycore CPUs	3X	Many simple cores (10s to 100s per chip), useful if software can be fine-grain parallel, difficult to maintain.
GPUs	5-30X	Good for data parallelism by merged threads (SIMD), High memory bandwidth, power hungry
FPGAs	5-30X	Most radical fully programmable option. Good for streaming/irregular parallelism. Power efficient but currently need to program in H/W languages.
Structured ASICS	20-100X	Lower-NRE ASICs with lower performance/efficiency. Includes domain-specific (programmable) accelerators.
Custom ASICs	> 100X	Highest efficiency. Highest NRE costs. Requires high volume. Good for functions in very widespread use that are stable for many years.

Specialized

More Efficient

Conventional programming

C/C++

CUDA

Alternative programming

Verilog

Can't change functionality

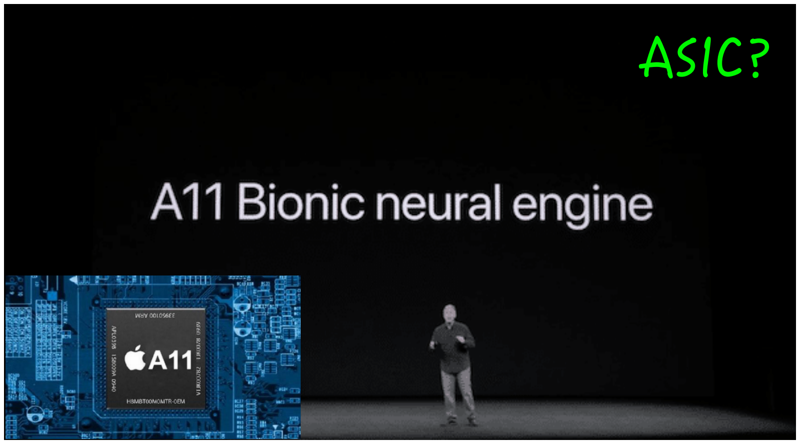
Verilog

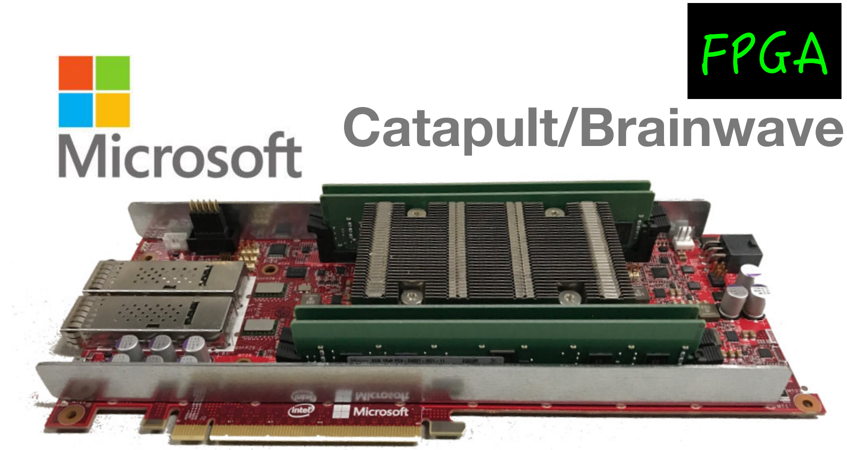
Power bill

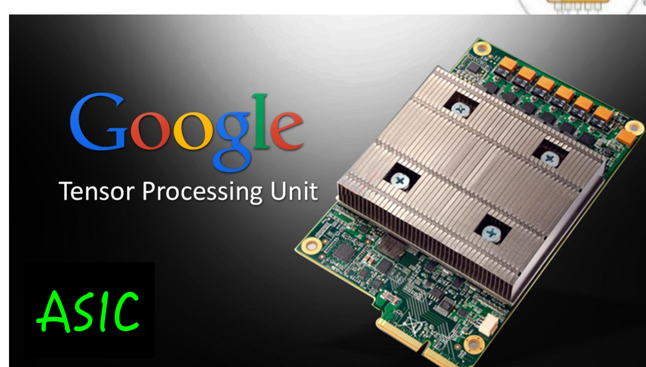
Software/Electrical engineer Salaries

#TRENDING IN INDUSTRY: CO-PROCESSORS

Specialized co-processor hardware for machine learning inference







INTEL® FPGA ACCELERATION HUB

The Intel® Xeon® Acceleration Stack for FPGAs is a robust framework enabling data center applications to leverage an FPGA's potential to increase

POSSIBLE SOLUTIONS FOR US

Computationally intensive: iterative algorithms such as track reconstruction

Option 1

re-write physics algorithms for new hardware

Language: OpenCL, OpenMP,TBB, HLS, …?

Hardware: FPGA, GPU

Option 2

re-cast physics problem as a machine learning problem

Language: C++, Python (TensorFlow, PyTorch,…)

Hardware: FPGA, GPU, ASIC

Example: tracking@HL-LHC:

Option 1: Parallelized and Vectorized Tracking Using Kalman Filters

Option 2: Recent work on tracking using Graph Networks

POSSIBLE SOLUTIONS FOR US

Option 1

re-write physics algorithms for new hardware

Language: OpenCL, OpenMP,TBB, HLS, …?

Hardware: FPGA, GPU

Option 2

re-cast physics problem as a machine learning problem

Language: C++, Python (TensorFlow, PyTorch,…)

Hardware: FPGA, GPU, ASIC

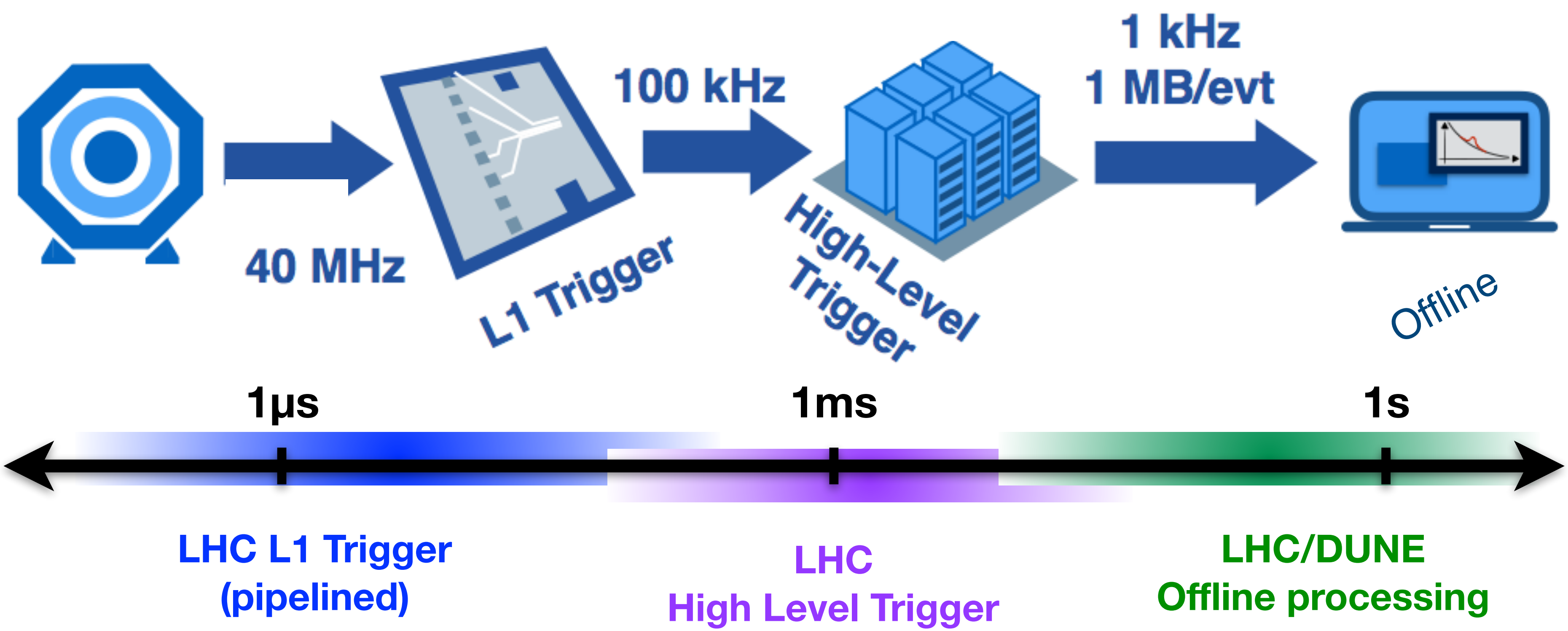
Advantage of option 2: recasting problem as machine learning problems (computing wise)

- Algorithms can universally be expressed as simple matrix multiplications computations
- Intrinsically parallelizable
- Follow industry trends in developing co-processors optimized for ML and speed the up the inference(sub-event level reconstruction such as tracking)

Proof of concept: Particle physics computing with Brainwave

Will explain this later with
pretty pictures
Picked this because of its
mature eco system

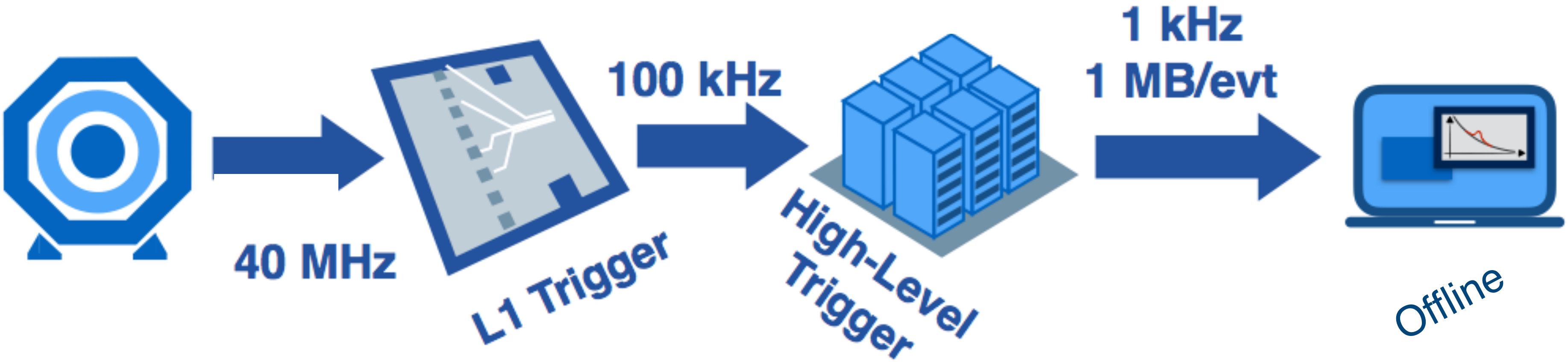
EVENT PROCESSING @ CMS EXPERIMENT



FPGAs/ASICs - high bandwidth low latency specialized compute hardware

Traditionally CPU, growing exploration into heterogeneous computing...
Brainwave!

EVENT PROCESSING @ CMS EXPERIMENT



LHC L1 Trigger (pipelined)

FPGAs/ASICs - high bandwidth low latency specialized compute hardware

LHC High Level Trigger

LHC/DUNE Offline processing

Traditionally CPU, growing exploration into heterogeneous computing…

Brainwave!

Two parallel talks this afternoon for L1/HLT applications:

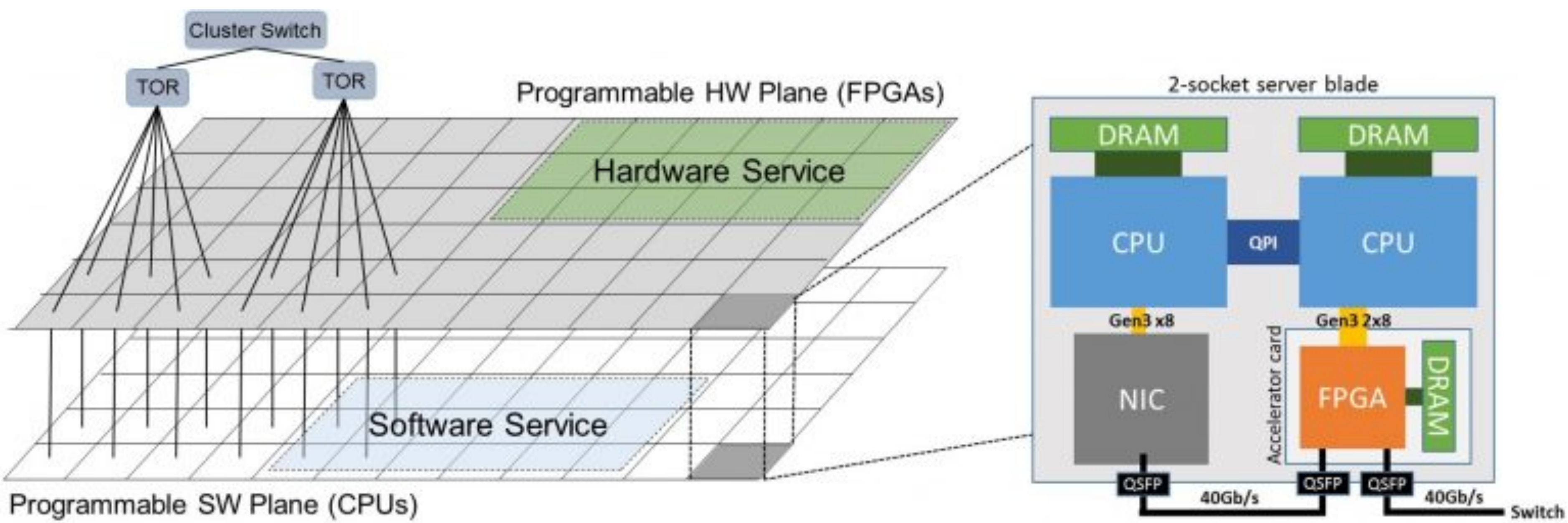
"DNN based algorithm for CMS Level-1 muon reconstruction" by Jia Low.

"Deep Machine Learning on FPGAs for L1 trigger and Data Acquisition" by Dylan Rankin

CO-PROCESSORS WITH WEB SERVICE: BRAINWAVE

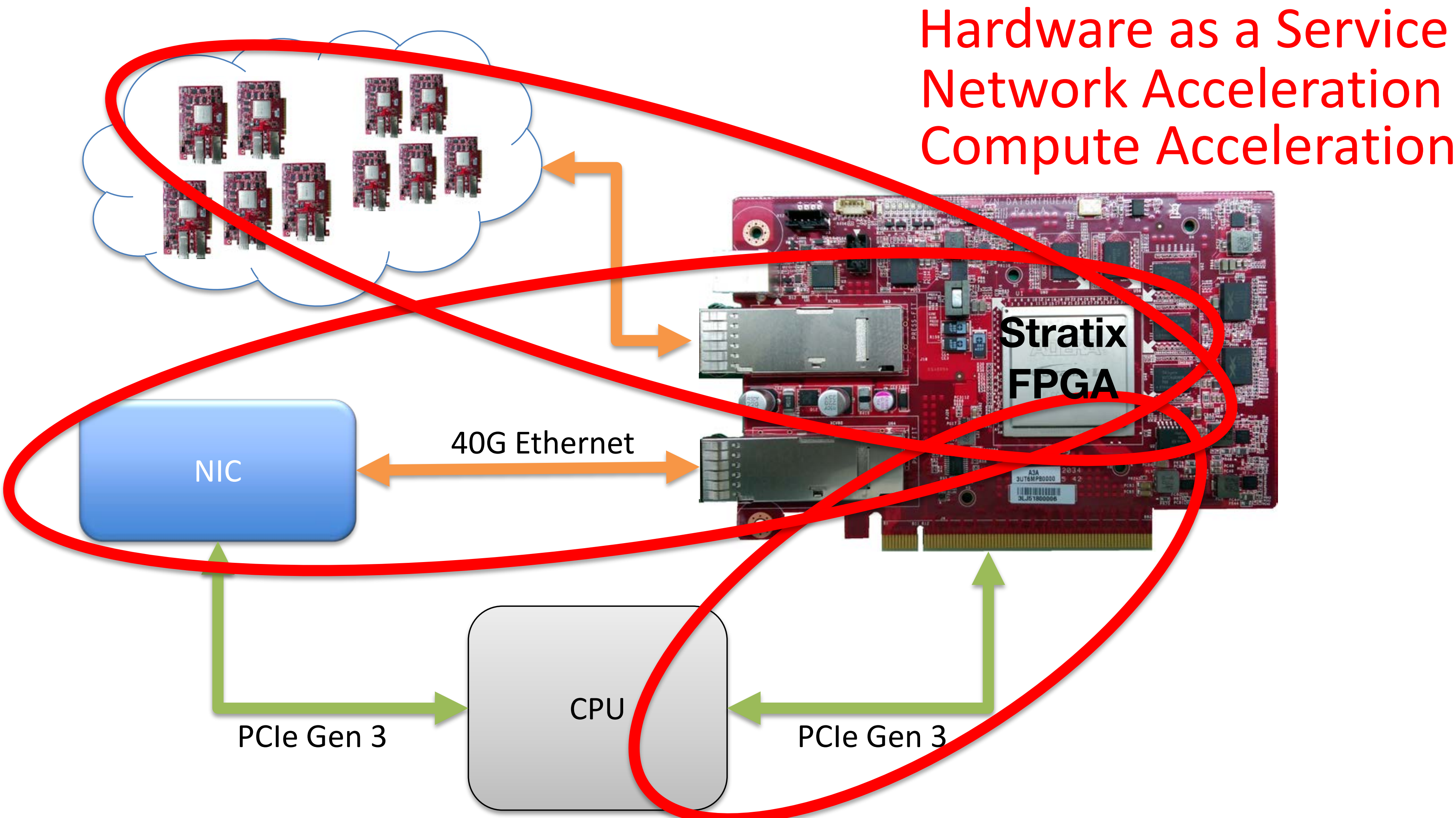
Even if co-processors are 100x faster, is it feasible to have every T1,T2,T3 computing farm buy specialized hardware?

No, but…
Interesting possibility for the HLT farm…



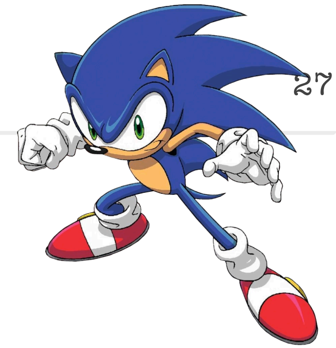
Offline solution: co-processors as a service

BUILDING BLOCKS: CATAPULT V2



For more on MS catapult: see talk by A. Putnam
https://www.dropbox.com/s/rvd06vp5ogguqxe/Catapult_2018_Fermilab_Public.pdf

PROOF OF CONCEPT: SONIC



Services for **O**ptimized **N**etwork **I**nference on **C**o-processors

PRELIMINARY RESULTS!

(work in progress)

FPGA-accelerated machine learning inference as a solution for particle physics computing challenges

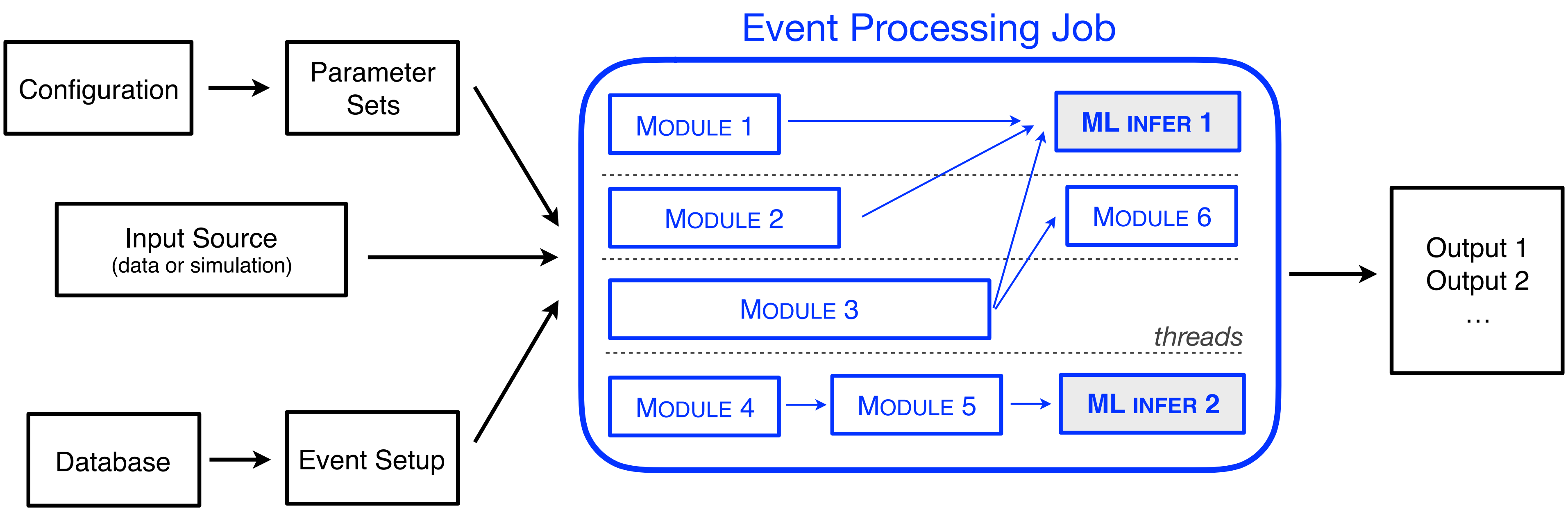
Javier Duarte, Burt Holzman, Ben Kreis, Mia Liu, Kevin Pedro, N.T., Aris Tsaris (FNAL)
Phil Harris, Dylan Rankin (MIT)

+ Doug Burger, Eric Chung, Andrew Putnam (MS research), Ted Way,MS, David Lee (MS Azure)

Question:
How do we integrate heterogeneous computing resources into the physics event data processing model?

PARTICLE PHYSICS COMPUTING MODEL

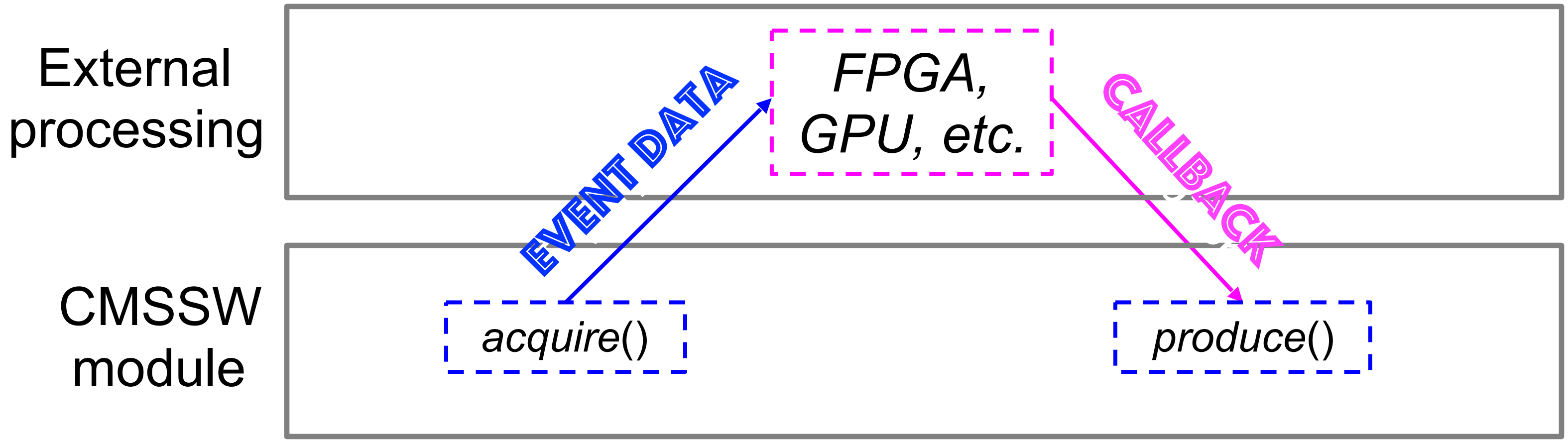
Our "unit" of analysis is at the event level, with complex interdependencies
Necessitates small "batch of a few" inferences



Accessing Heterogeneous Resources

Implemented New CMSSW feature called ExternalWork:

- ○ Asynchronous task-based processing



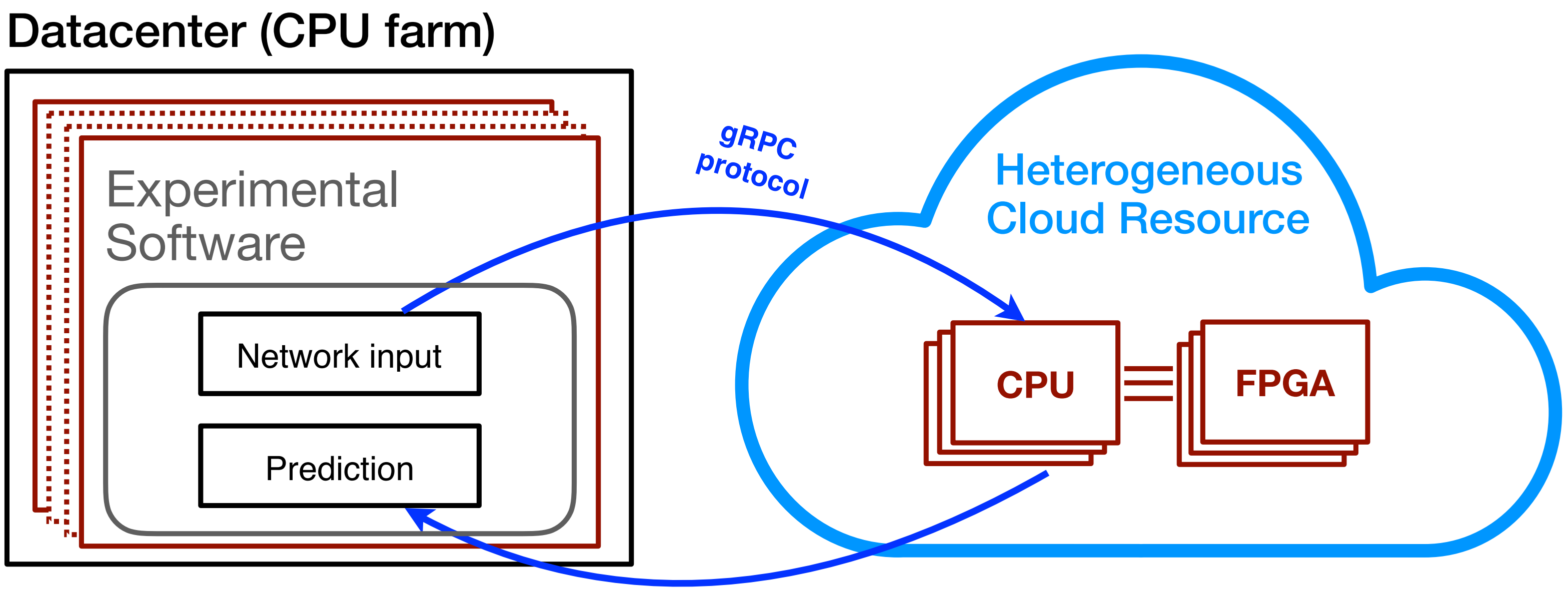
- ○ **Non-blocking**: schedule other tasks while waiting for external processing

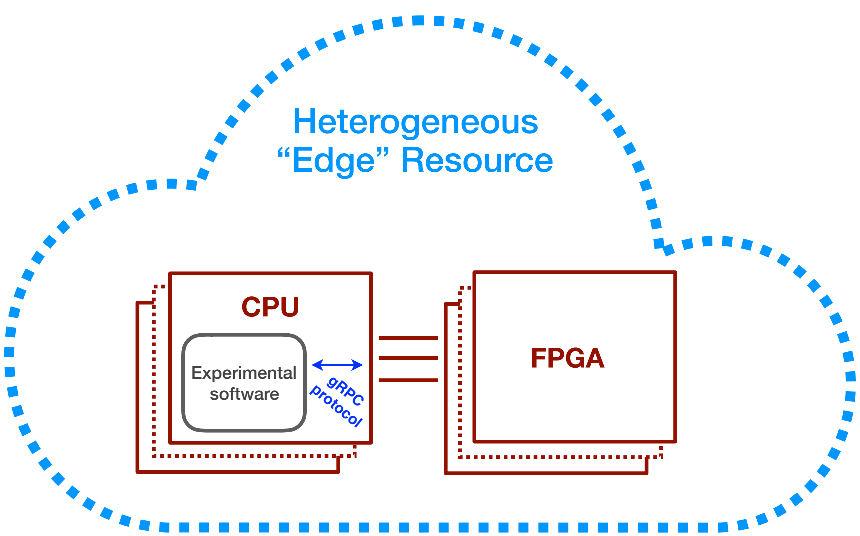
Can be used with GPUs, FPGAs, cloud, …

➢**Now demonstrated to work with Microsoft Brainwave!**

More details on external work module: [Kevin Pedro's talk at CHEP]

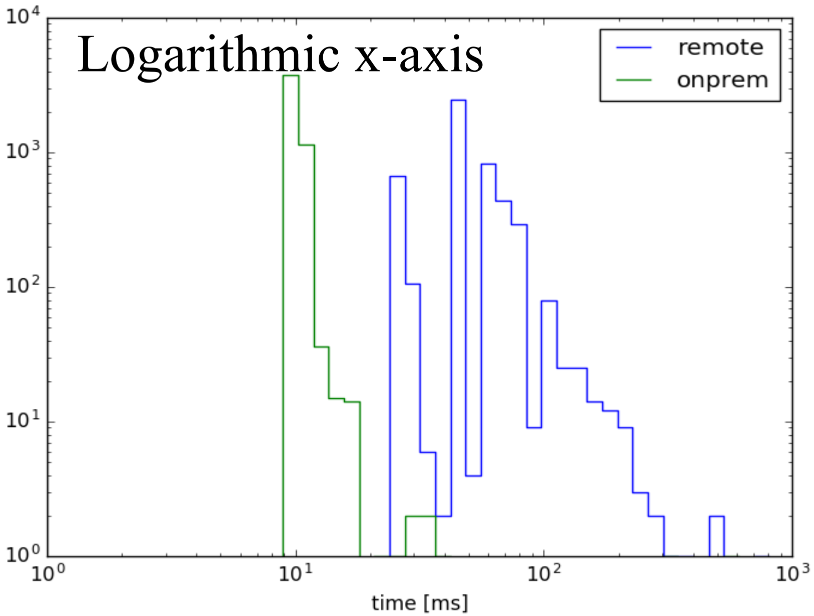
CLOUD VS EDGE

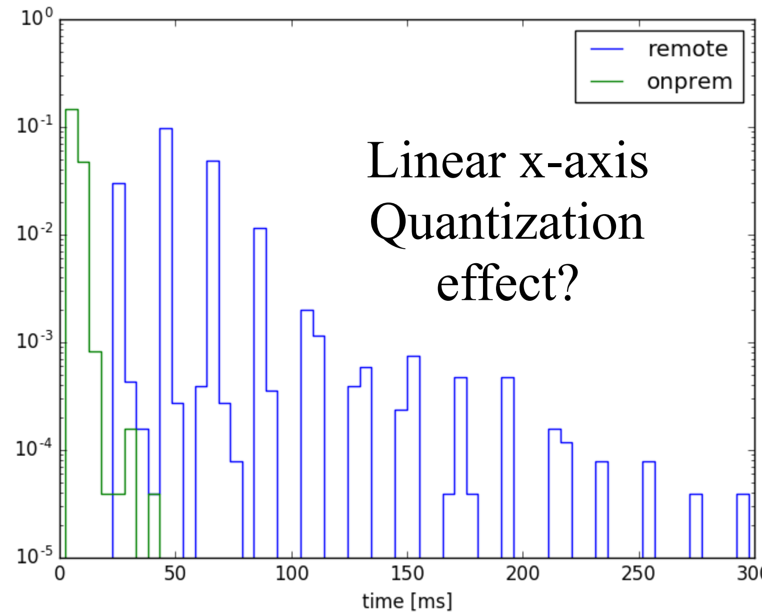




- Cloud service has latency
- Run CMSSW on Azure cloud machine → simulate local installation of FPGAs ("on-prem" or "edge")
- Provides test of "HLT-like" performance

TESTING SONIC





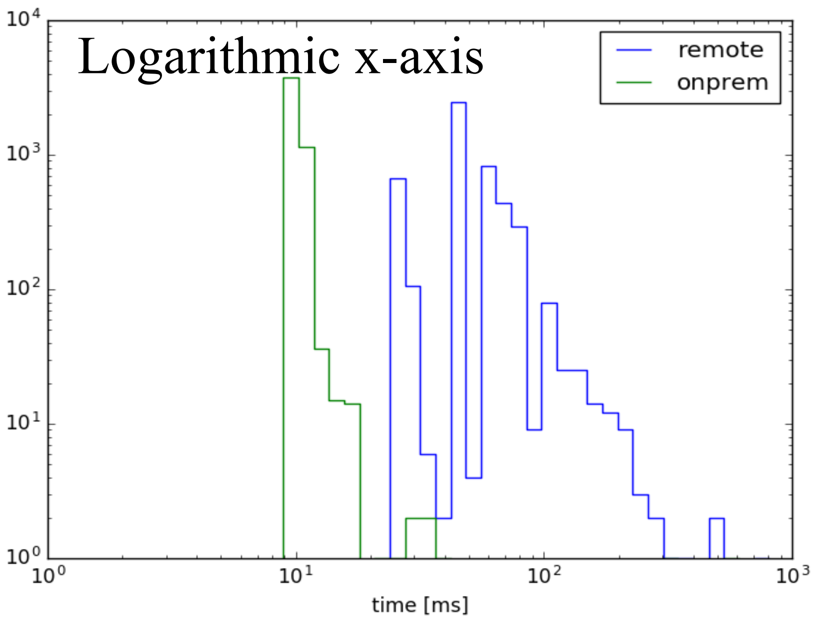
Good performance in initial tests

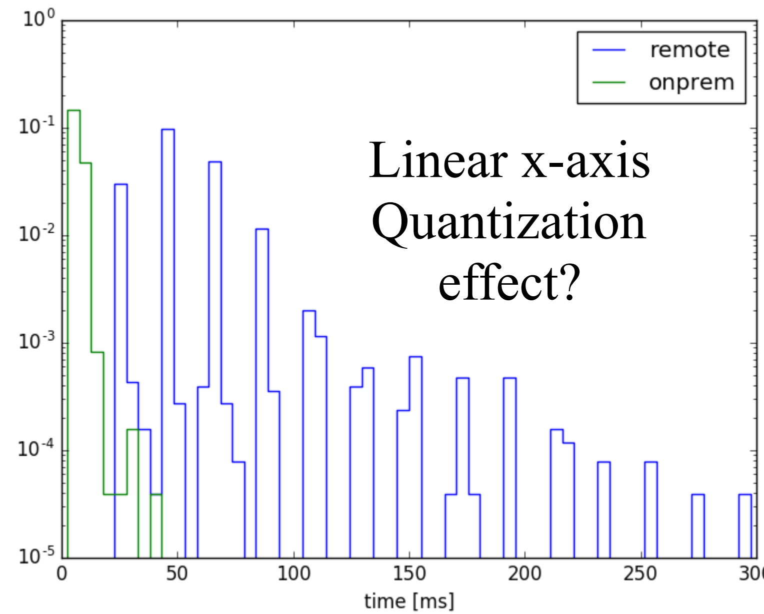
- "remote": cmslpc @ FNAL to Azure (VA), ‹time› = 56 ms
- "onprem": run CMSSW on Azure VM, ‹time› = 10 ms
  (~2 ms on FPGA, rest is classifying and I/O)

TRAVEL LATENCY?

With network switches? May be about right :)

Testing SONIC



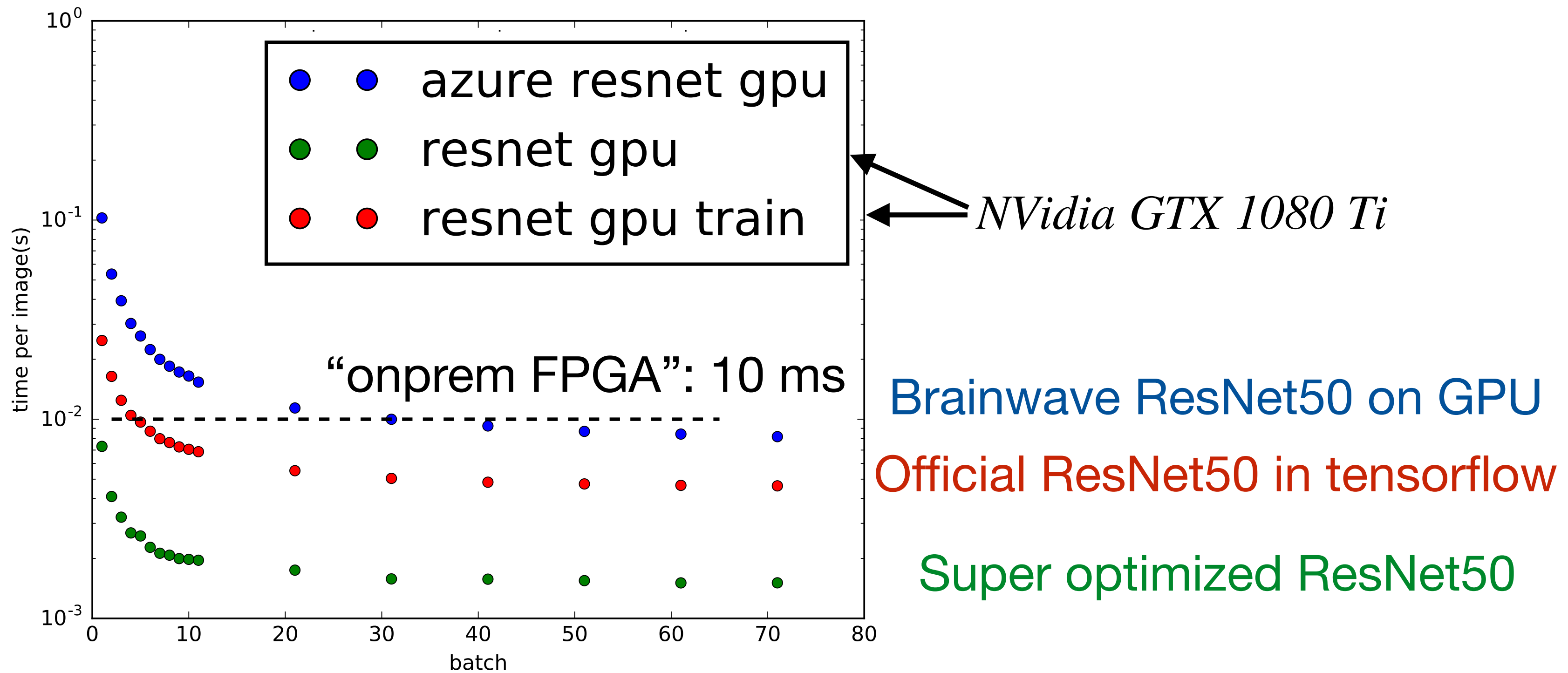


Good performance in initial tests

- "remote": cmslpc @ FNAL to Azure (VA), ‹time› = 56 ms
- "onprem": run CMSSW on Azure VM, ‹time› = 10 ms
  (~2 ms on FPGA, rest is classifying and I/O)

Type	Hardware	Mean inference time	Setup
CPU	Xeon 2.6 GHz, 1 core	1.75 seconds	`CMSSW, TF v1.06`
CPU	i7 3.6 GHz, 1 core	500 ms	standalone python, `TF v1.10`
CPU	i7 3.6 GHz, 8 core	200 ms	standalone python, `TF v1.10`

Other hardware options



Not so straightforward to compare against other hardware, the whole chain matters: pipelined inputs, IO bandwidth (PCIe), special instruction sets, etc. General findings:

GPUs: O(~100 ms), for batch-1 input

To explore: Google TPUs, AWS/Xilinx FPGAs, Intel/Altera FPGAs

HLT/OFFLINE OUTLOOK

Exploring the use of FPGA co-processors (MS Brainwave) for ML acceleration as an “off-the-shelf” computing paradigm for particle physics

- Deploying cloud accelerators as a service **fits the particle physics computing model in a non-disruptive way**
  - For large computing tasks (Resnet-50), **there is a ~(4/10/100)x benefit over CPU-only computations**
  - Could be used for neutrino experiments **~today!**
- “Edge” compute option as an HLT solution?

Outlook and further studies

To explore: Google TPUs, AWS/Xilinx FPGAs, Intel/Altera FPGAs

Important to benchmark different platforms to understand our options/projections.

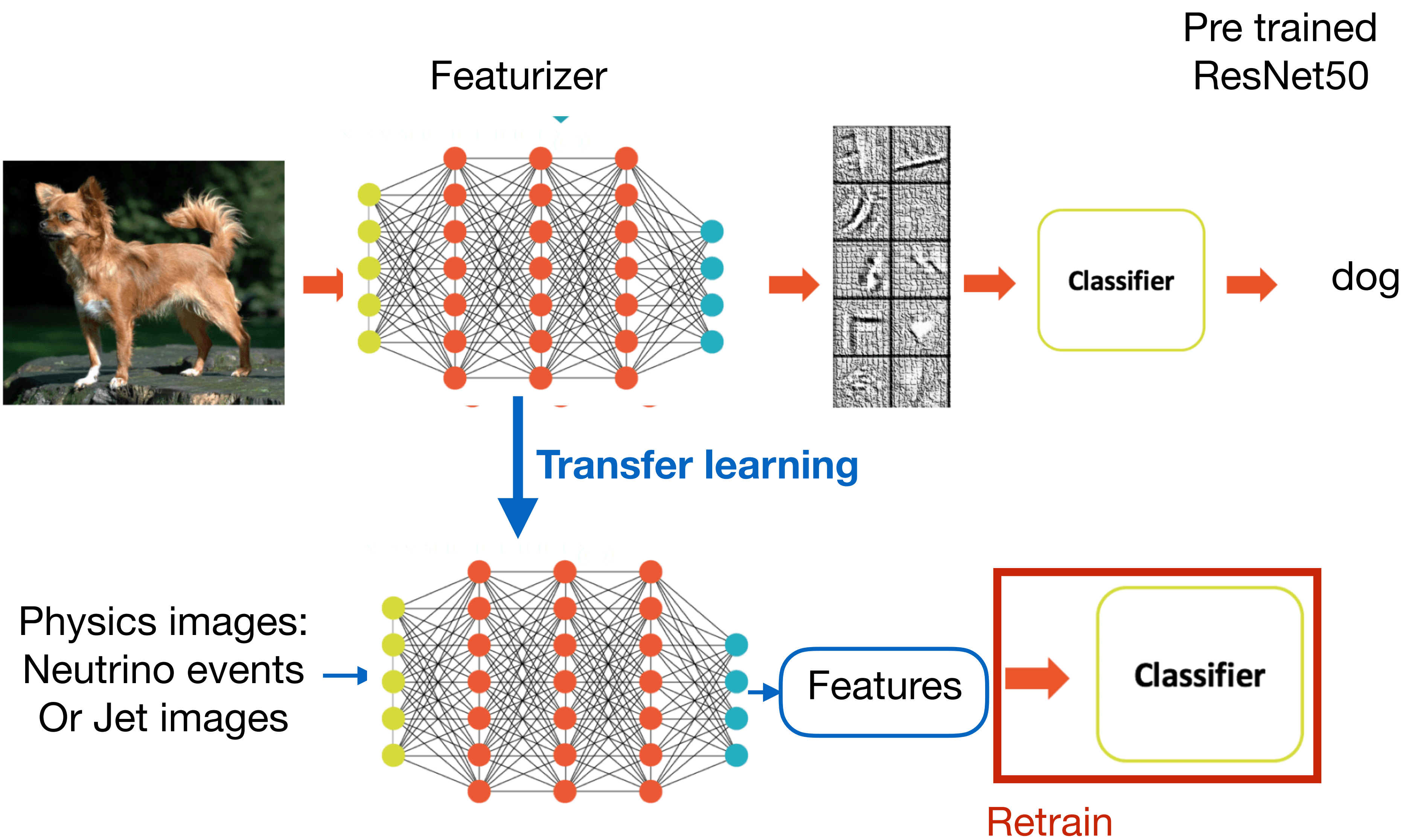
Need to understand scaling (not too worried about this)

What’s the costing model?

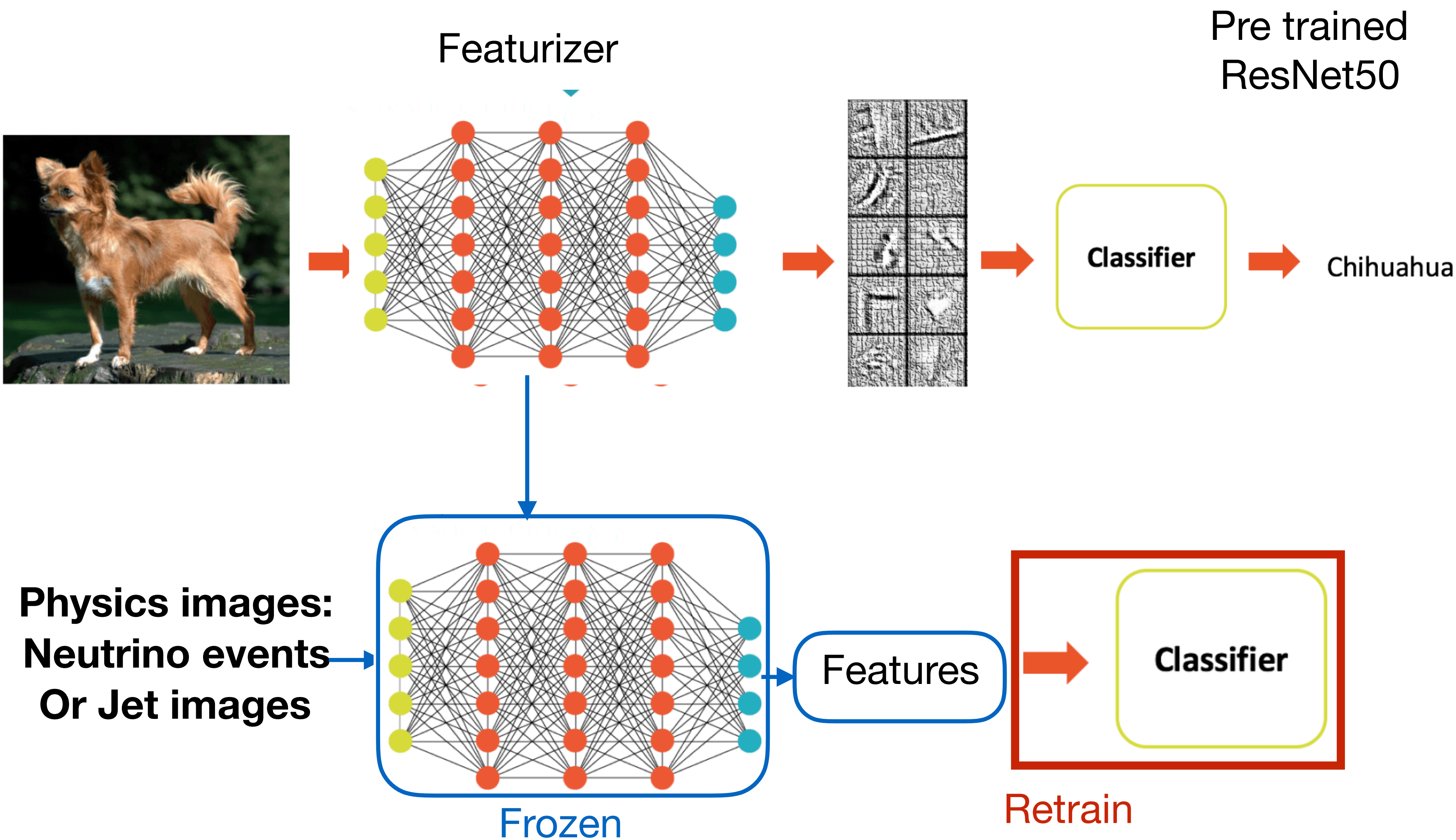
Relies on continued development of ML algorithms for difficult physics problems (simulation/reconstruction)

Physics case: Event classification in NOvA

Available Models and services on Brainwave



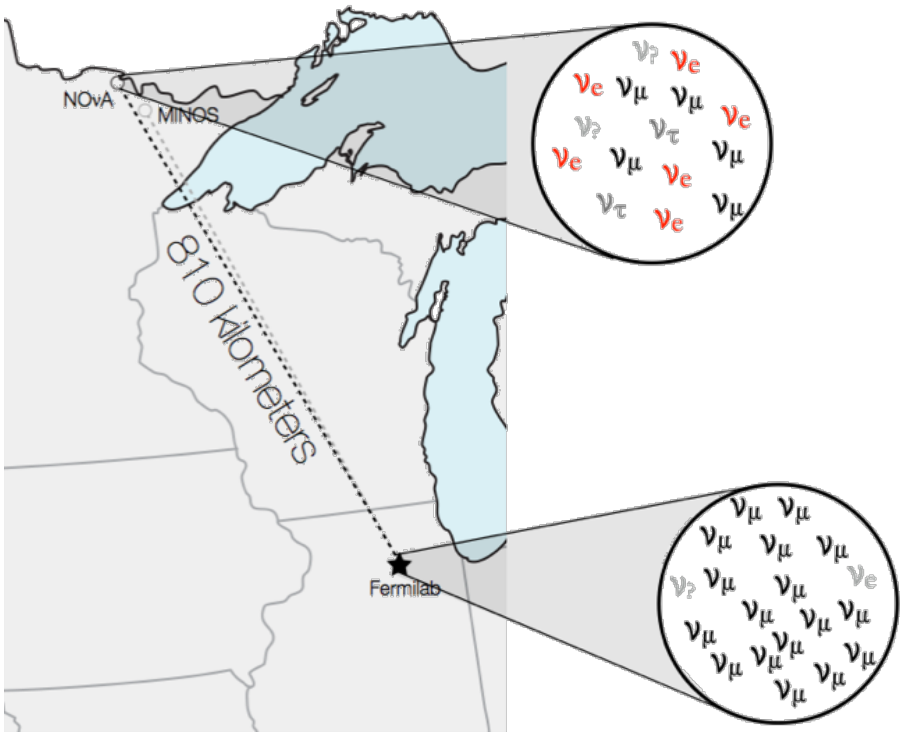
AVAILABLE MODELS AND SERVICES ON BRAINWAVE



New feature: fine-tune the weights in featurizer too! Will be included in final results

Other models became available recently: VGG etc

THE NOvA EXPERIMENT:



NuMI: Neutrinos at the Main Injector

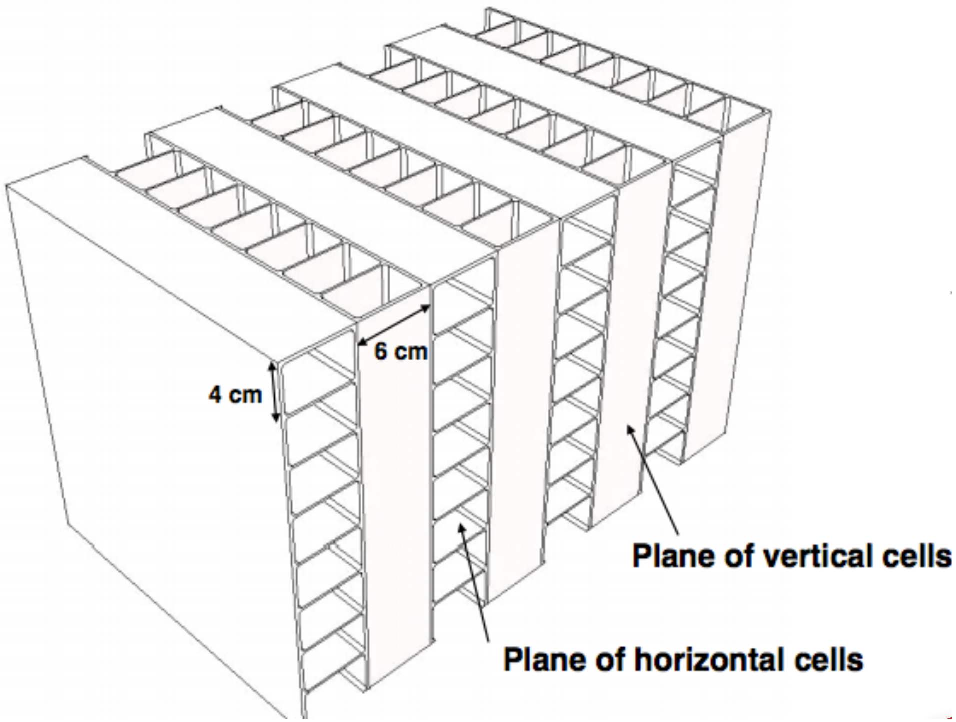
Long-baseline (anti-)neutrino oscillation experiment

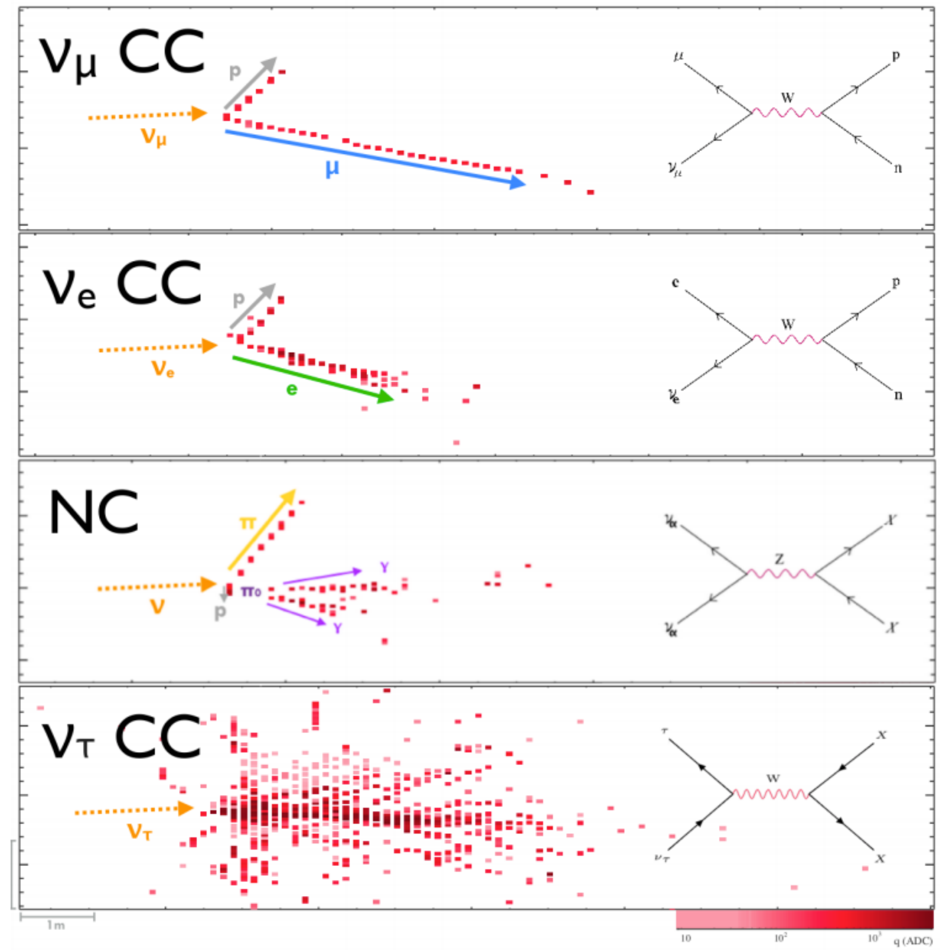
Two functionally identical detectors, optimized for ve identification

Primary goal:
measurement of neutrino oscillations via vμ→ve

Other goals include:
Searches for sterile neutrinos
Neutrino cross sections
Supernova neutrinos Cosmic ray physics

EVENT TOPOLOGY



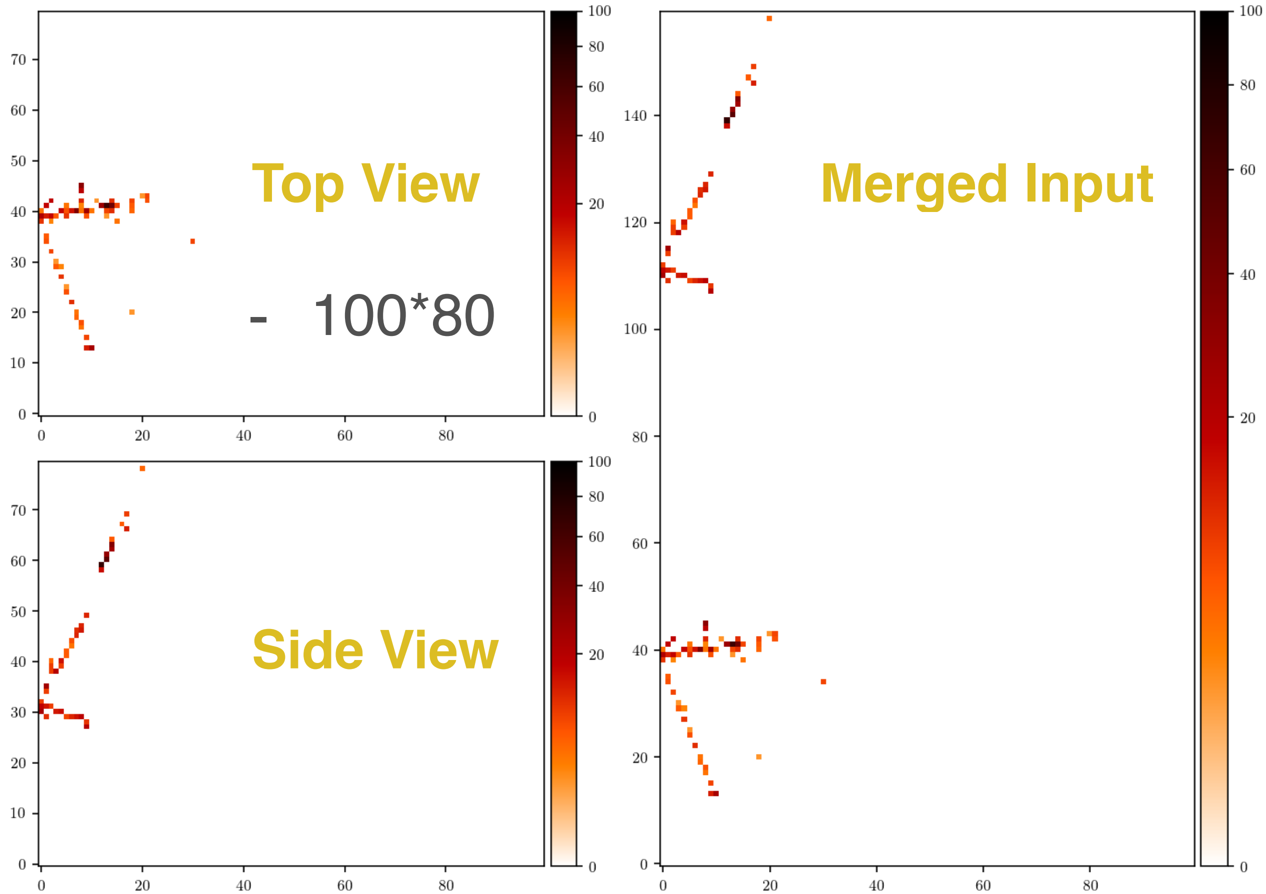


3D reconstruction

NOvA:Neutrino Flavor Classification

Training Setup

- training/testing: 500K/150K
- Pre-trained ResNet-50 model on image net.

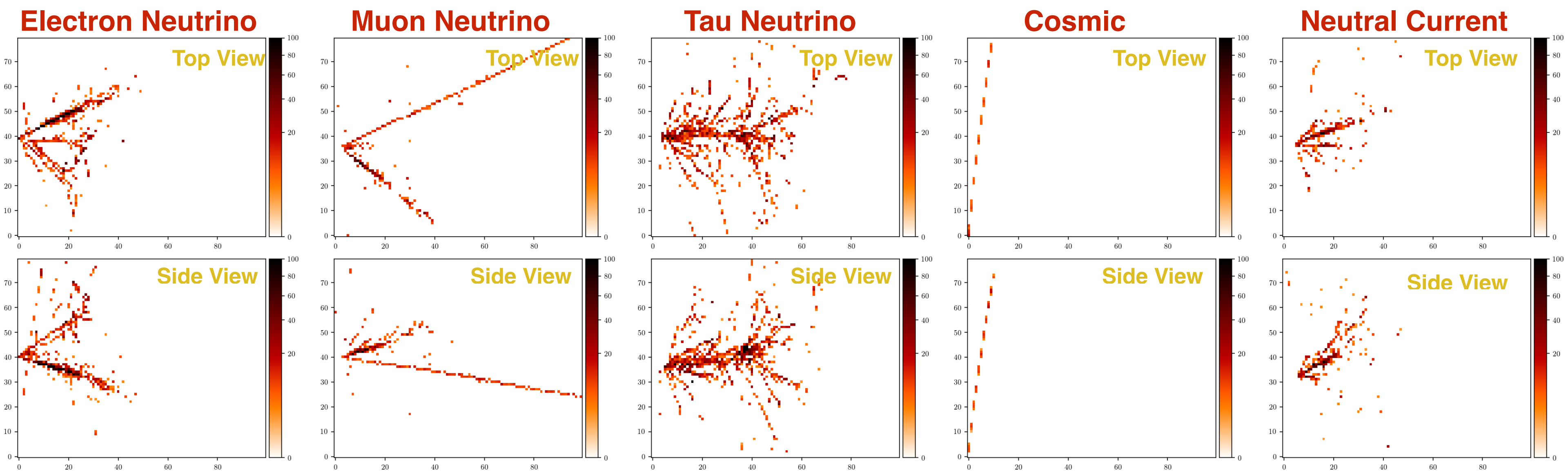


5 labels:

- Muon neutrino
- Electron neutrino
- Tau neutrino
- Neutral Current
- Cosmic

- Merged image scaled to resolution of 224*224 using Bilinear Interpolation from TF, to be fed into ResNet50

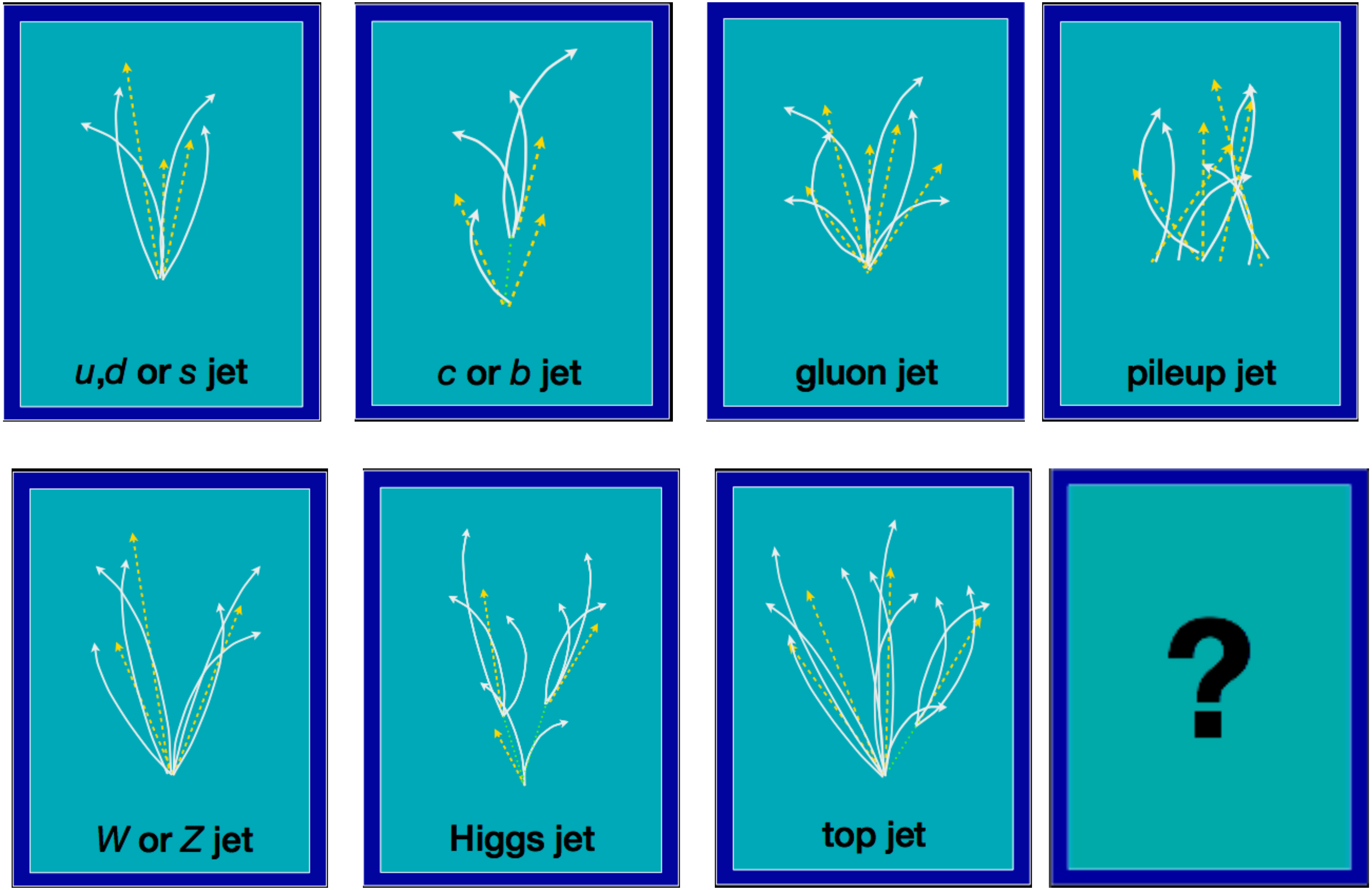
Events with Probability larger than 0.9



Events identified with more that 0.9 probability by the ResNet-50 network.
Color represents energy deposit

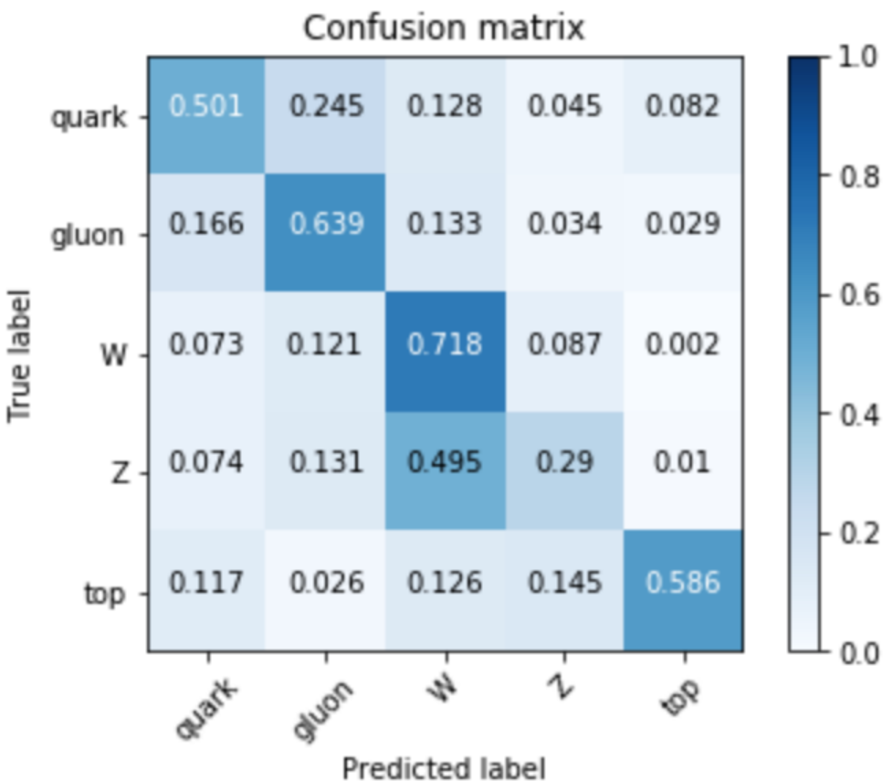
Physics cases: jet substructure

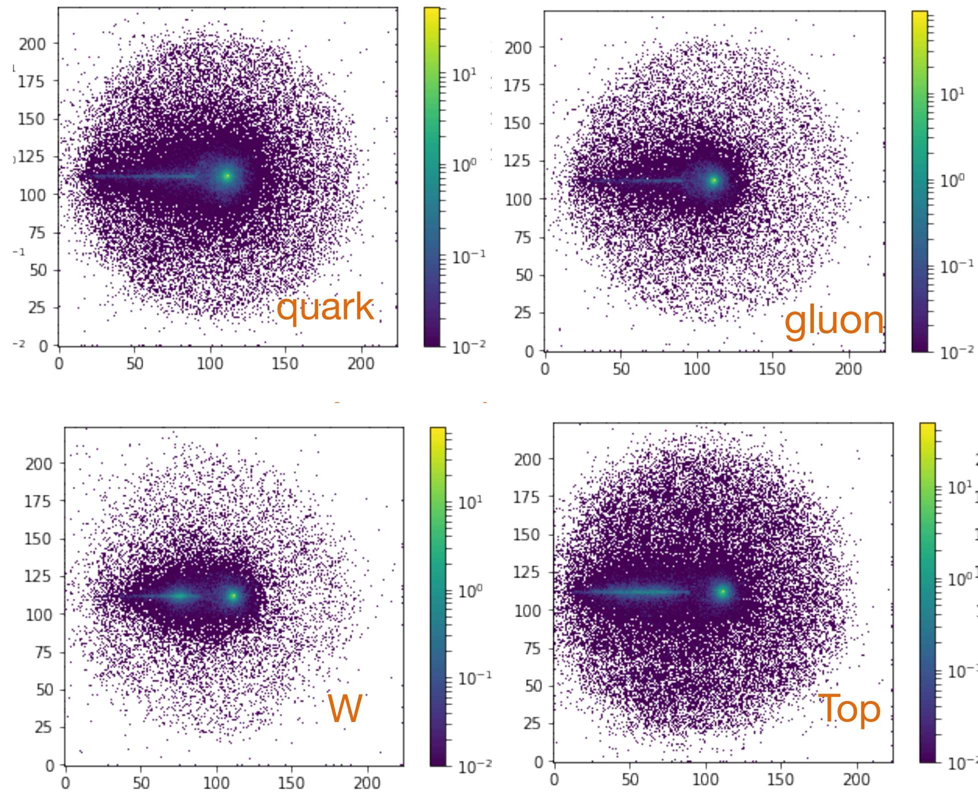
JET SUBSTRUCTURE



TRAINING STATUS

Generator level AK8 jets: quark/gluon/W/Z/top, density map of the pt of jet constituents.





Averaged over 1000 images

Outlook and next steps

TAKEAWAYS

Computing challenges in big data: More complex detectors and sophisticated algorithms, large datasets.

We follow the industry trend in exploring specialized hardware (co-processers) as ML acceleration options.

Started with Microsoft brainwave, **demonstrated FPGAs are a promising option to accelerate neural network inference:**

o *Can achieve (at least) order of magnitude improvement over CPU*

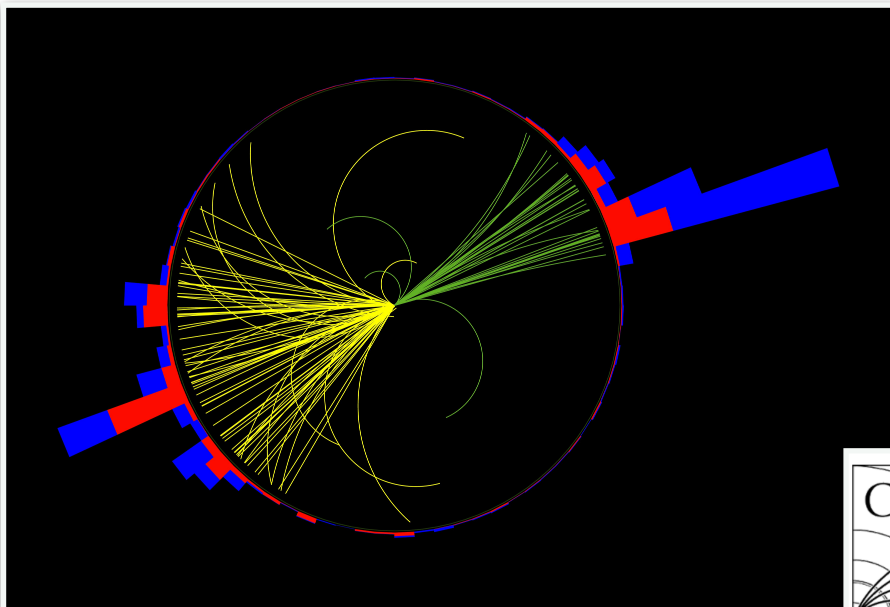
o *Better fit for CMS event-level computing model (vs. GPUs which require batching for efficiency)*

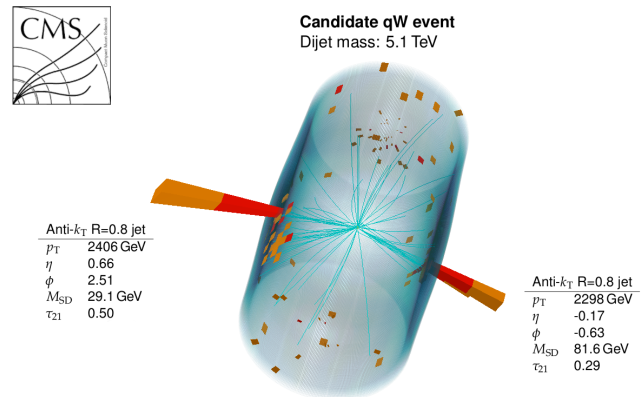
o *Physics cases: Nova event classification& jet substructure using ResNet50 on brainwave.*

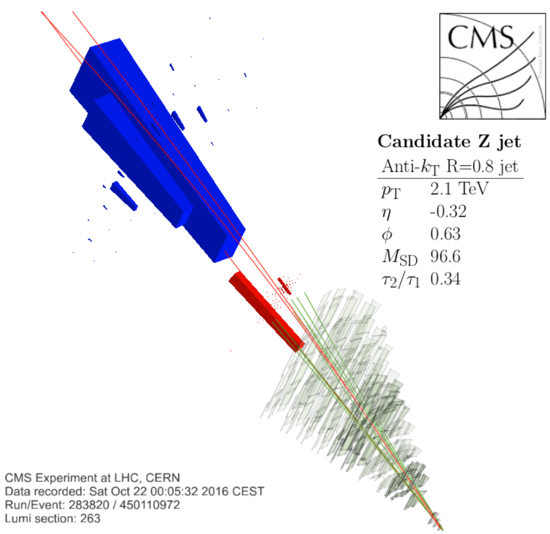
Proof of concept, more studies to follow

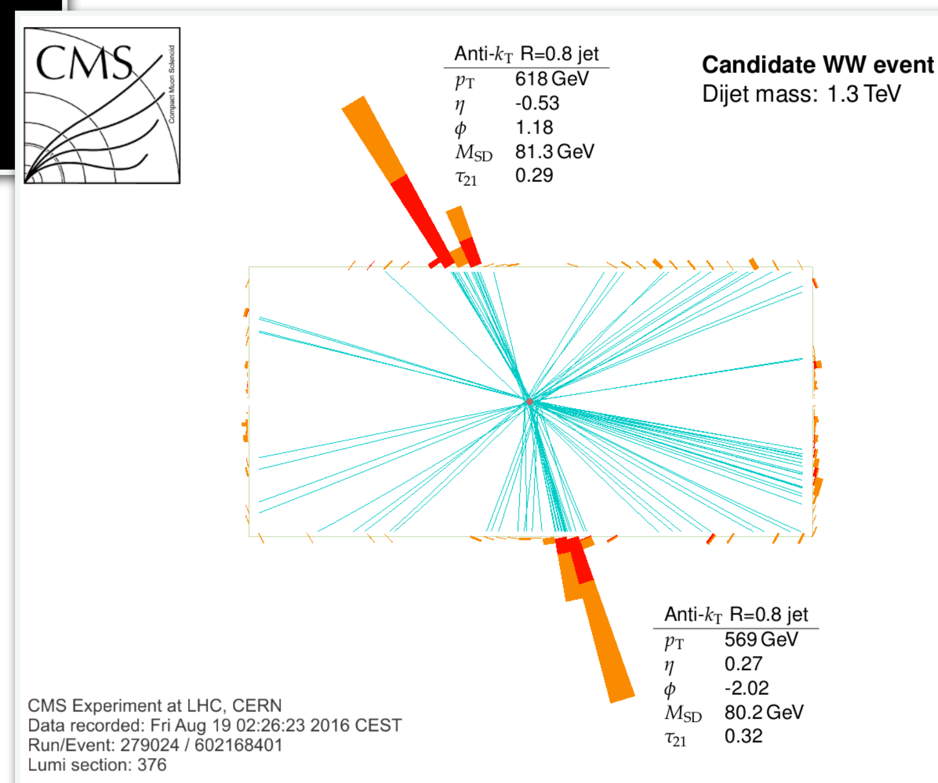
BACKUP

PICTURES




CMS
Candidate qW event
Dijet mass: 5.1 TeV
Anti-k_T R=0.8 jet
p_T 2406 GeV
η 0.66
ϕ 2.51
M_{SD} 29.1 GeV
τ_{21} 0.50
Anti-k_T R=0.8 jet
p_T 2298 GeV
η -0.17
ϕ -0.63
M_{SD} 81.6 GeV
τ_{21} 0.29


CMS
Candidate Z jet
Anti-k_T R=0.8 jet
p_T 2.1 TeV
η -0.32
ϕ 0.63
M_{SD} 96.6
τ_2/τ_1 0.34
CMS Experiment at LHC, CERN
Data recorded: Sat Oct 22 00:05:32 2016 CEST
Run/Event: 283820 / 450110972
Lumi section: 263


CMS
Compact Muon Solenoid
Candidate WW event
Dijet mass: 1.3 TeV
Anti-k_T R=0.8 jet
p_T 618 GeV
η -0.53
ϕ 1.18
M_{SD} 81.3 GeV
τ_{21} 0.29
Anti-k_T R=0.8 jet
p_T 569 GeV
η 0.27
ϕ -2.02
M_{SD} 80.2 GeV
τ_{21} 0.32
CMS Experiment at LHC, CERN
Data recorded: Fri Aug 19 02:26:23 2016 CEST
Run/Event: 279024 / 602168401
Lumi section: 376

FURTHER NEXT STEPS

Model customization

Whenever we talk about this — people are generally positive but always ask when we can put our own networks on the FPGAs

Is it something we can work with you on? Not just CNNs, but Graph NNs, LSTMs, etc…

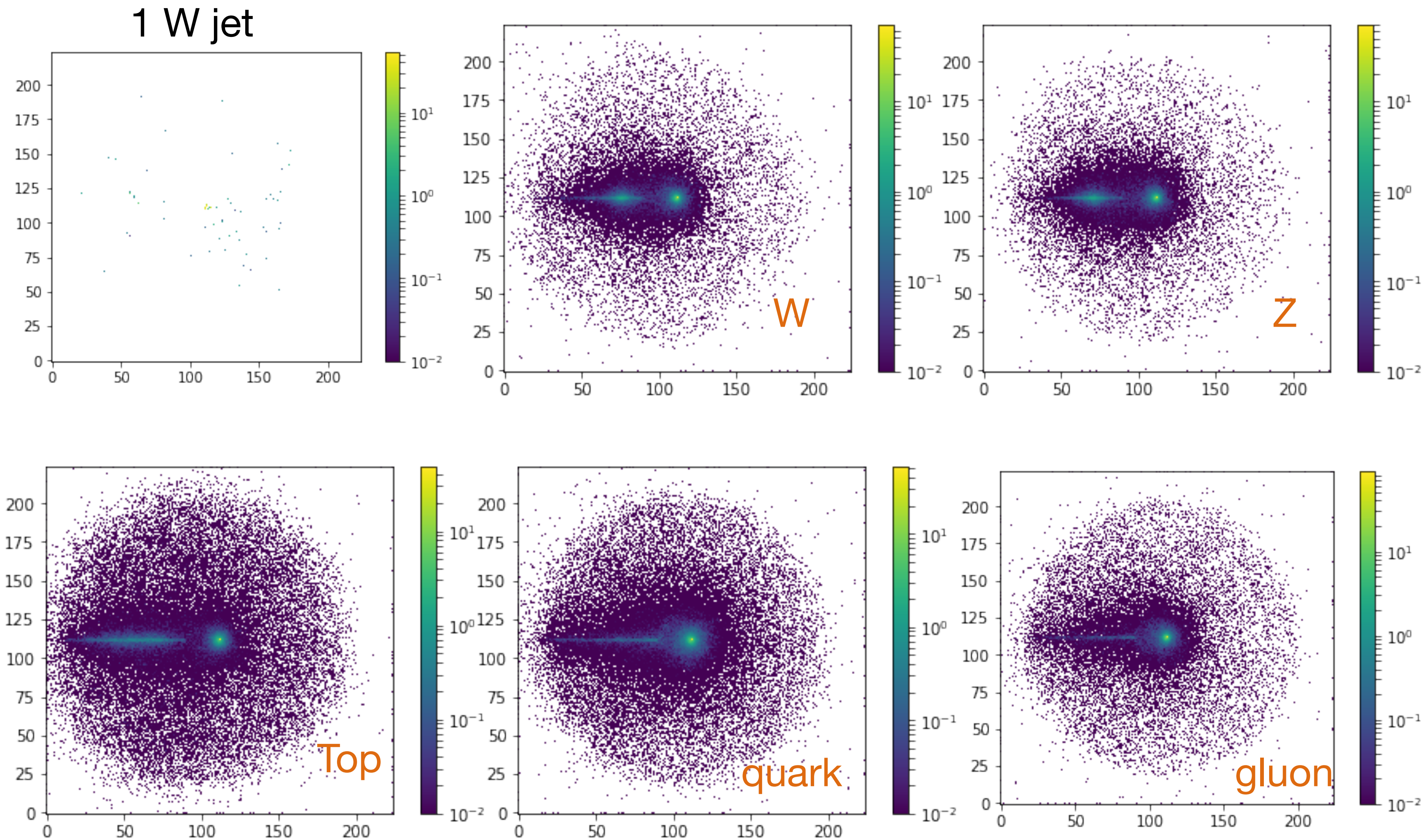
On-prem HLT-like demonstration

An “edge” offering has been brought up a few times — this is something, with necessary infrastructure, we’re interested in pursuing if possible as a demonstration of the trigger (on-prem, real-time) capabilities

Scaling up

We should try to demonstrate running on N (>>1) CPUs and M (>1) services to understand how to scale services. This will give us an idea of cost scaling as well.

JET IMAGES



Averaged over 1000 images

The CMS Software

CMSSW:

- Hosted on [GitHub]
- ~6 million lines of code
- Handles simulation, raw data processing, reconstruction, analysis

Event-based processing model:

- Load event data into memory
- Numerous modules process parts of event, output new products

Parallelism:

- Multiple events in flight → *streams*
- Multiple modules running simultaneously → *threads*
  - Task-based multithreading using Intel Thread Building Blocks

WLCG: Worldwide LHC Computing Grid

- Network of computing clusters at labs, universities, etc.
- Mostly commodity hardware

SONIC IN CMSSW

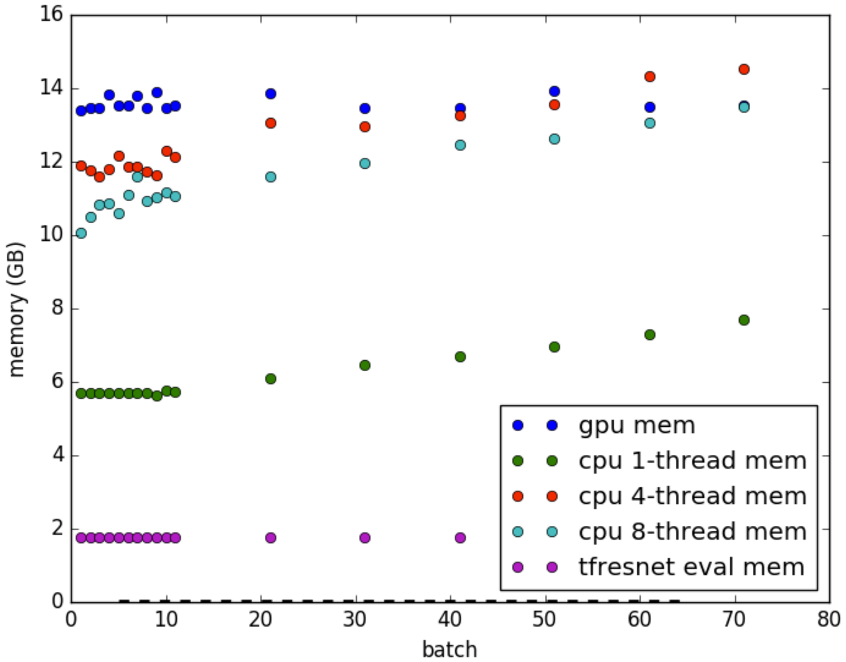
Services for Optimized Network Inference on Coprocessors

Demonstration in Microsoft Brainwave:

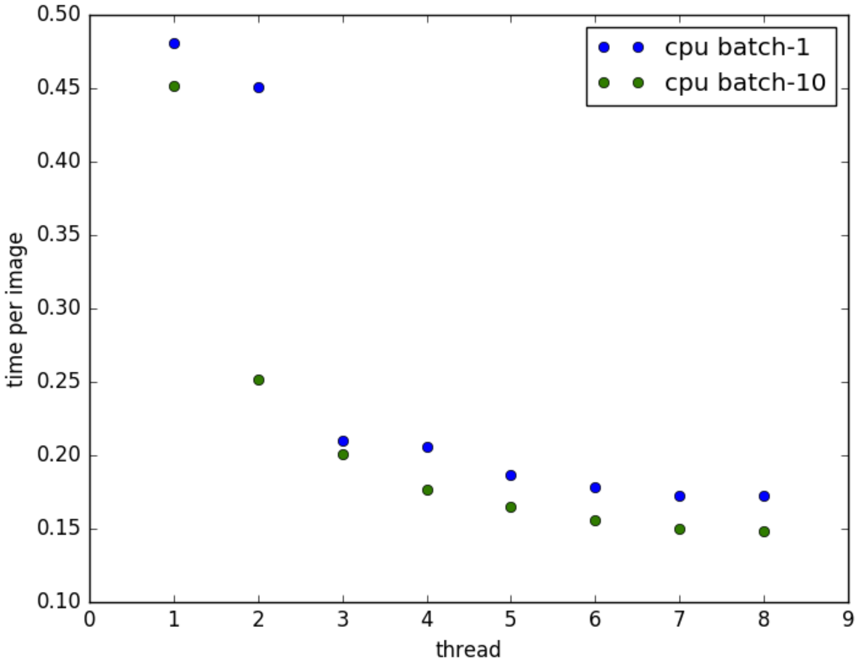
- Create “image” from jet constituents, process with ResNet50
  - Much larger than custom HEP networks (so far)
- Send to Microsoft Brainwave FPGA using gRPC w/ TensorFlow (protobuf)
- FPGA processes one image at a time → no batching needed to be efficient
- Use ExternalWork mechanism
  - gRPC C++ API lacks a callback interface (currently)
    - ➢wait for gRPC return in lightweight std::thread

SonicCMS repository on GitHub

MEMORY AND THREADS



Memory usage in different configurations

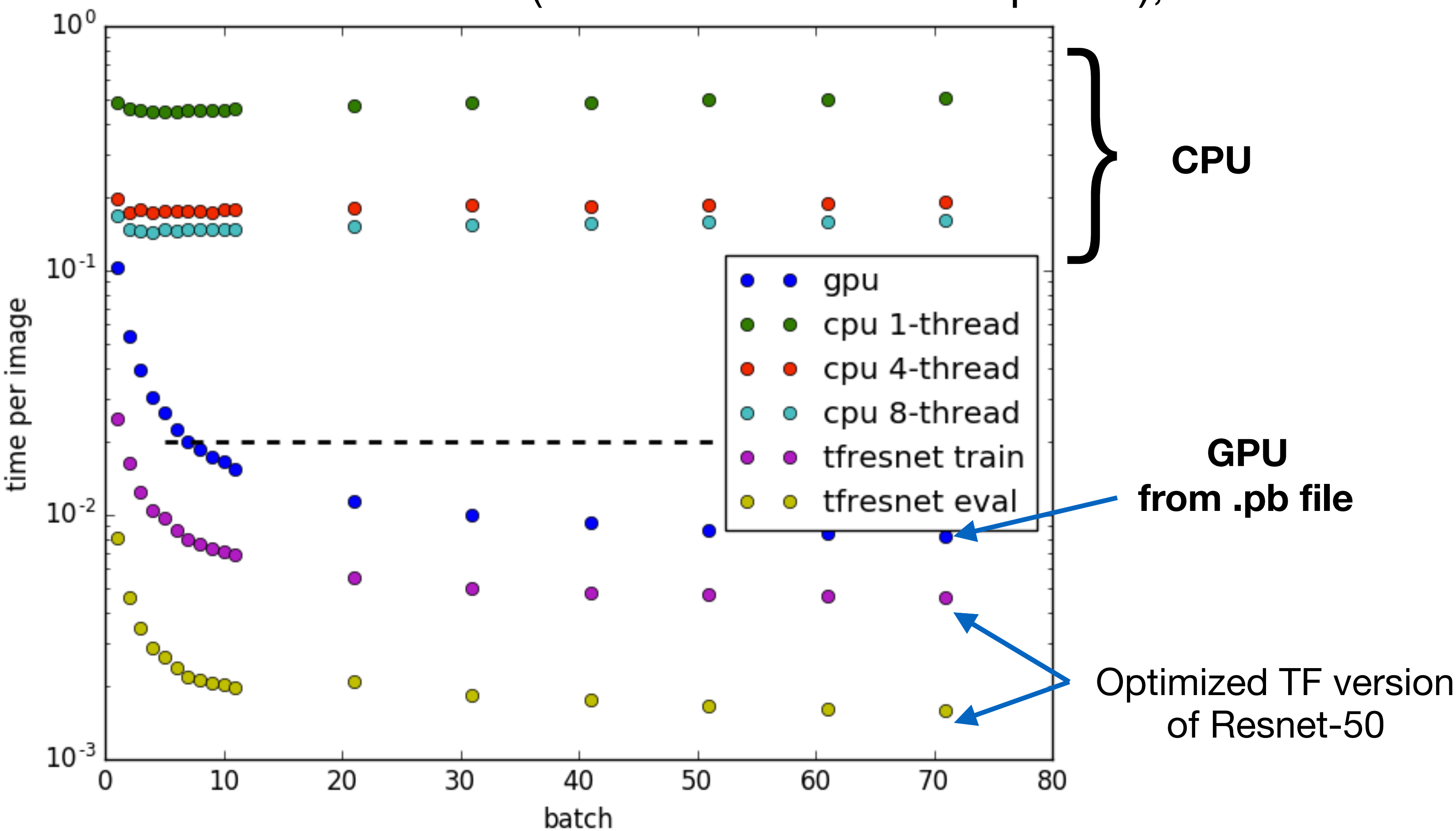


Latency versus number of threads

COMPARISON TO GPU/CPU: UPDATE

Benchmark Nvidia GTX 1080, Intel i7 3.6 GHz

Pure inference time (load time is 5 min for .pb file), TF v1.10



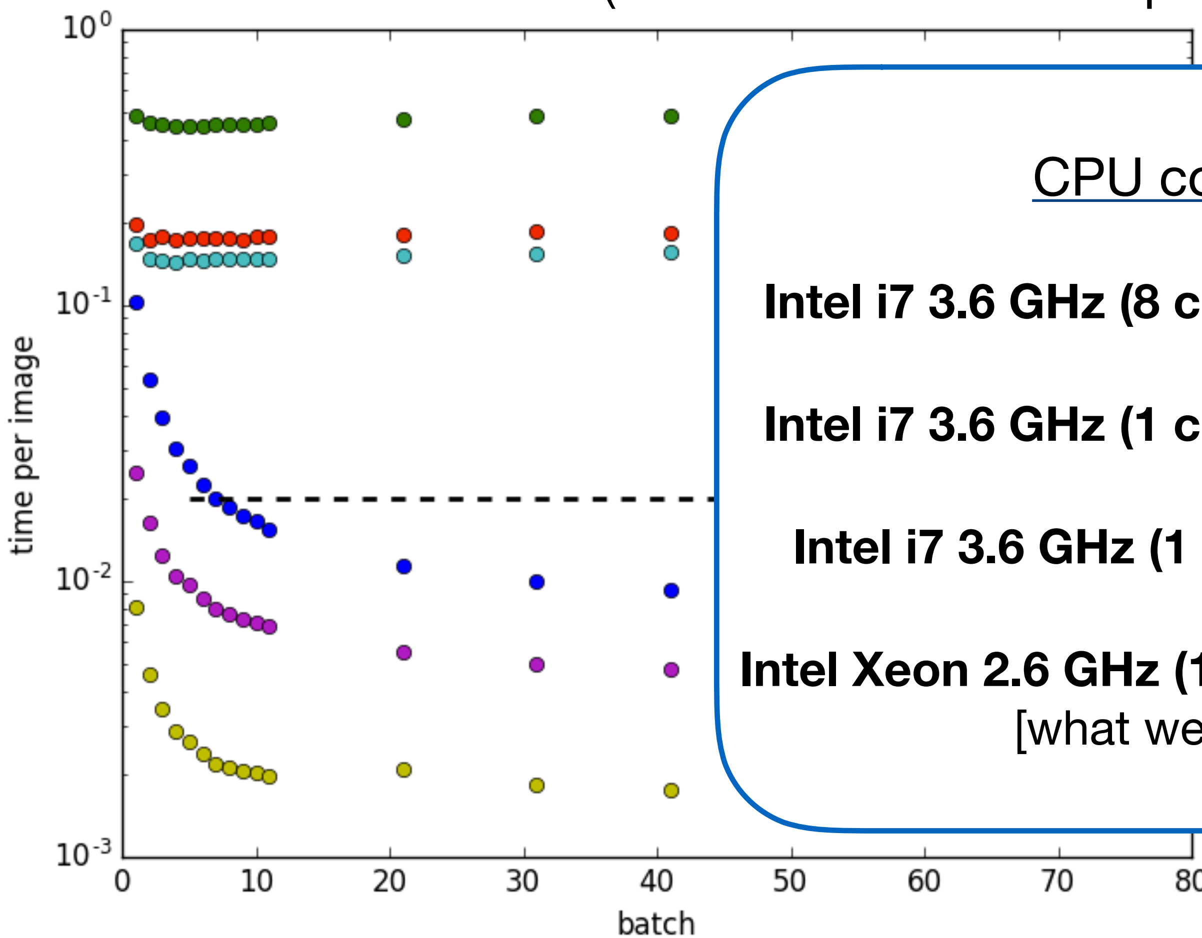
Full enqueuing with random inputs,

Large memory usage (12 Gb) with .pb input

COMPARISON TO GPU/CPU

Benchmark Nvidia GTX 1080, Intel i7 3.6 GHz

Pure inference time (load time is 5 min for .pb file), TF v1.10



CPU comparison:

Intel i7 3.6 GHz (8 core, TF v1.10) ~ 180 ms

Intel i7 3.6 GHz (1 core, TF v1.10) ~ 500 ms

Intel i7 3.6 GHz (1 core, TF v1.06) ~ 1.2 s

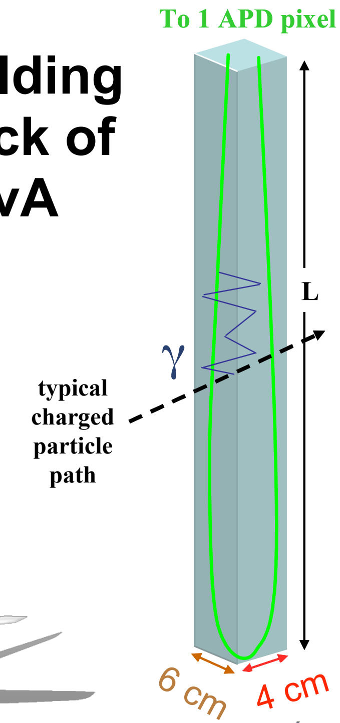
Intel Xeon 2.6 GHz (1 core, TF v1.06) ~ 1.75 s
[what we are running]

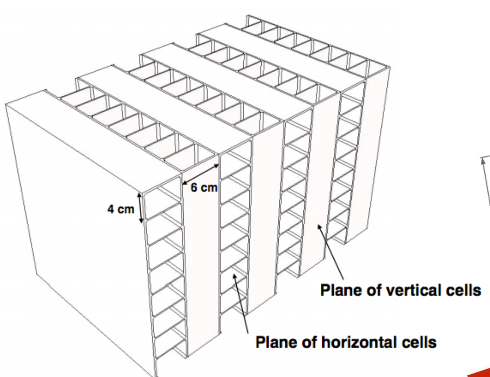
Full enqueuing with random inputs,
Large memory usage (12 Gb) with .pb input

NOvA Detectors

- Highly segmented low Z tracking calorimeter.
- Cells are filled wit liquid scintillator
  - wave shifting fiber readout.
- 65% active by volume
- Detection with avalanche photo diodes.
- Alternating X/Y planar geometry: **3D reconstruction**

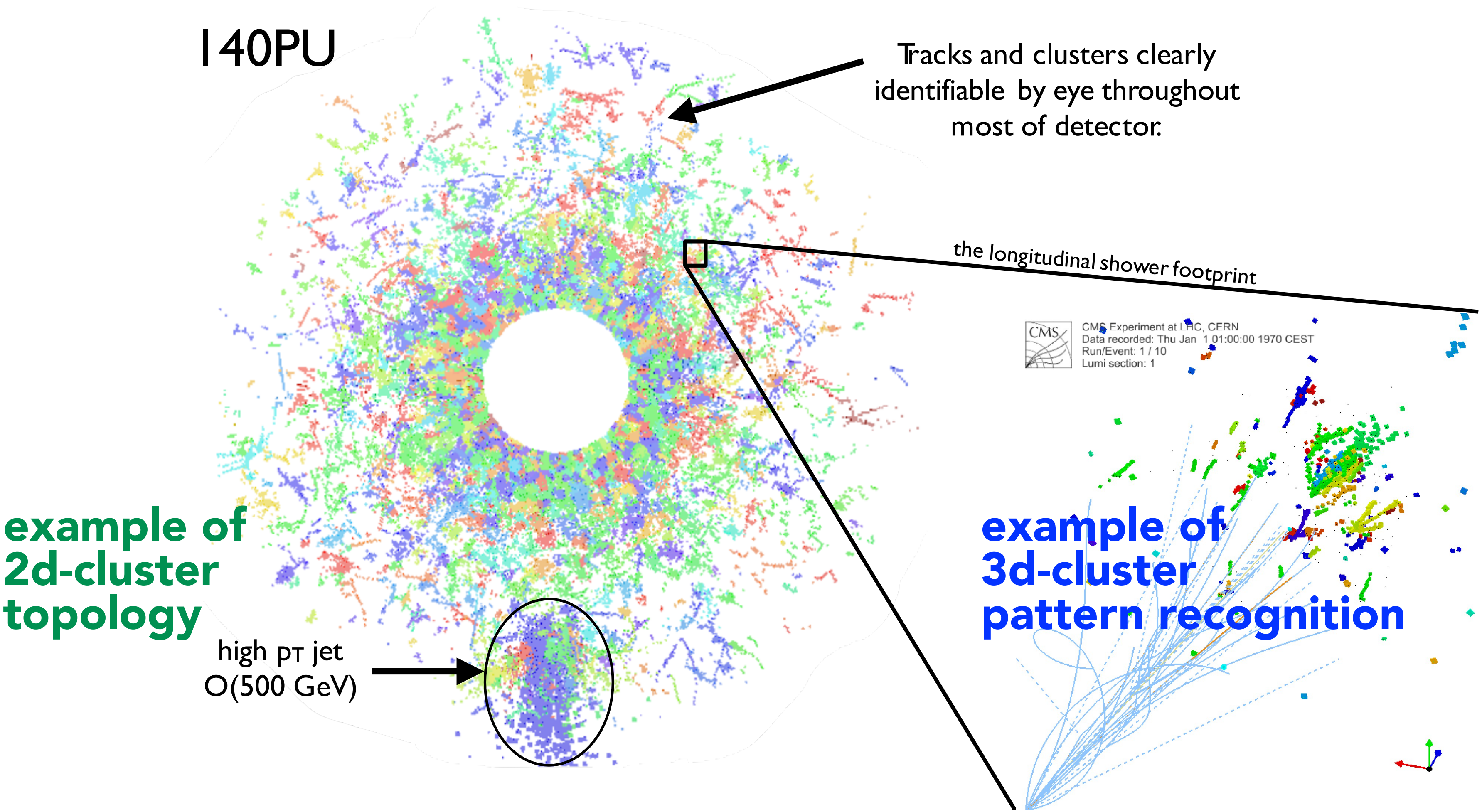
Building block of NOvA





Far Detector, on surface
14 kTons
896 readout planes
344,064 pixels

ANOTHER EXAMPLE: CMS HGC



CMS PARTY@2016

Privacy issue: not the focus today but probably deserves a plenary talk in some other conferences…